2756

#2 SSK 10/25/00

862.C1922

PATENT APPLICATION

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Application of:
NOBORU HAMADA

Appln. No.: 09/588,672

Filed: June 6, 2000

For: NETWORK DEVICE MANAGING APPARATUS AND METHOD

Examiner: Not Yet Known

Group Art Unit: 2756

October 19, 2000

Commissioner for Patents
Washington, D.C. 20231

CLAIM TO PRIORITY

Sir:

Applicant hereby claims priority under the International Convention and all rights to which he is entitled under 35 U.S.C. § 119 based upon the following Japanese Priority Application:

11-165569 filed on June 11, 1999

A certified copy of the priority document, together with an English translation of the first page of the same, containing the filing data, is enclosed.

CFM 1922 US
09/588,692

日 本 国 特 許 庁
PATENT OFFICE
JAPANESE GOVERNMENT

別紙添付の書類に記載されている事項は下記の出願書類に記載されている事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed with this Office.

出 願 年 月 日
Date of Application: 1999年 6月11日

出 願 番 号
Application Number: 平成11年特許願第165569号

出 願 人
Applicant (s): キヤノン株式会社

CERTIFIED COPY OF PRIORITY DOCUMENT

2000年 6月29日

特許庁長官
Commissioner,
Patent Office 近藤隆彦

出証番号 出証特2000-3050745

【書類名】　特許願

【整理番号】　3933107

【提出日】　平成11年 6月11日

【あて先】　特許庁長官殿

【国際特許分類】　G06F 15/00

【発明の名称】　ネットワークデバイス管理装置及び方法

【請求項の数】　10

【発明者】

【住所又は居所】　東京都大田区下丸子３丁目３０番２号　キヤノン株式会社内

【氏名】　浜田　昇

【特許出願人】

【識別番号】　000001007

【氏名又は名称】　キヤノン株式会社

【代理人】

【識別番号】　100076428

【弁理士】

【氏名又は名称】　大塚　康徳

【電話番号】　03-5276-3241

【選任した代理人】

【識別番号】　100093908

【弁理士】

【氏名又は名称】　松本　研一

【電話番号】　03-5276-3241

【選任した代理人】

【識別番号】　100101306

【弁理士】

【氏名又は名称】　丸山　幸雄

【電話番号】　03-5276-3241

【手数料の表示】

【予納台帳番号】　003458

【納付金額】　21,000円

【提出物件の目録】

【物件名】　明細書　1

【物件名】　図面　1

【物件名】　要約書　1

【包括委任状番号】　9704672

【プルーフの要否】　要

【書類名】　明細書

【発明の名称】　ネットワークデバイス管理装置及び方法

【特許請求の範囲】

【請求項１】　起動後に少なくとも１度はネットワーク管理用パケットをブロードキャストするデバイスが接続されたネットワークの管理装置であって、

パケットを受信するパケット受信手段と、

前記パケット受信手段で受信したパケットが、ネットワーク管理用のパケットであるか判定するパケット判定手段と、

前記パケット判定手段でネットワーク管理用パケットであると判断された場合、そのパケットから、そのパケットを送信したデバイスのアドレスを取得するデバイスアドレス取得手段と、

前記デバイスアドレスを登録するデバイスアドレス登録手段と

を備えることを特徴とするネットワークデバイス管理装置。

【請求項２】　起動後に少なくとも１度はネットワーク管理用パケットをブロードキャストするデバイスが接続されたネットワークの管理装置であって、

パケットを受信するパケット受信手段と、

前記パケット受信手段で受信したパケットが、ネットワーク管理用のパケットであるか判定するパケット判定手段と、

前記パケット判定手段でネットワーク管理用パケットであると判断された場合、そのパケットを送信したデバイスに対して、そのデバイスが監視の対象となるデバイスであることを確認するための確認パケットを送信する送信手段と、

前記監視パケットに対する応答に基づいて、その応答を返したデバイスの状態をアドレスと共に登録するデバイスアドレス登録手段と

を備えることを特徴とするネットワークデバイス管理装置。

【請求項３】　前記応答は、その応答元のデバイスが管理対象のデバイスであれば正常応答であり、該デバイスの稼働状態を示す情報を含むことを特徴とする請求項２に記載のネットワークデバイス管理装置。

【請求項４】　前記送信手段は、プリンタの状態を取得するための確認パケットを送信し、前記アドレス登録手段は、確認パケットに対する応答が正常応答

であれば、応答元のデバイスのアドレスと共に、そのデバイスがプリンタであることを登録することを特徴とする請求項２に記載のネットワークデバイス管理装置。

【請求項５】　起動後に少なくとも１度はネットワーク管理用パケットをブロードキャストするデバイスが接続されたネットワークの管理方法であって、

パケットを受信するパケット受信工程と、

前記パケット受信工程で受信したパケットが、ネットワーク管理用のパケットであるか判定するパケット判定工程と、

前記パケット判定工程でネットワーク管理用パケットであると判断された場合、そのパケットから、そのパケットを送信したデバイスのアドレスを取得するデバイスアドレス取得工程と、

前記デバイスアドレスを登録するデバイスアドレス登録工程と

を備えることを特徴とするネットワークデバイス管理方法。

【請求項６】　起動後に少なくとも１度はネットワーク管理用パケットをブロードキャストするデバイスが接続されたネットワークの管理方法であって、

パケットを受信するパケット受信工程と、

前記パケット受信工程で受信したパケットが、ネットワーク管理用のパケットであるか判定するパケット判定工程と、

前記パケット判定工程でネットワーク管理用パケットであると判断された場合、そのパケットを送信したデバイスに対して、そのデバイスが監視の対象となるデバイスであることを確認するための確認パケットを送信する送信工程と、

前記監視パケットに対する応答に基づいて、その応答を返したデバイスの状態をアドレスと共に登録するデバイスアドレス登録工程と

を備えることを特徴とするネットワークデバイス管理方法。

【請求項７】　前記応答は、その応答元のデバイスが管理対象のデバイスであれば正常応答であり、該デバイスの稼働状態を示す情報を含むことを特徴とする請求項６に記載のネットワークデバイス管理方法。

【請求項８】　前記送信工程は、プリンタの状態を取得するための確認パケットを送信し、前記アドレス登録工程は、確認パケットに対する応答が正常応答

であれば、応答元のデバイスのアドレスと共に、そのデバイスがプリンタであることを登録することを特徴とする請求項６に記載のネットワークデバイス管理方法。

【請求項９】　コンピュータにより、

パケットを受信するパケット受信手段と、

前記パケット受信手段で受信したパケットが、ネットワーク管理用のパケットであるか判定するパケット判定手段と、

前記パケット判定手段でネットワーク管理用パケットであると判断された場合、そのパケットから、そのパケットを送信したデバイスのアドレスを取得するデバイスアドレス取得手段と、

前記デバイスアドレスを登録するデバイスアドレス登録手段と

を実現するためのコンピュータプログラムを格納するコンピュータ可読の記憶媒体。

【請求項１０】　コンピュータにより、

パケットを受信するパケット受信手段と、

前記パケット受信手段で受信したパケットが、ネットワーク管理用のパケットであるか判定するパケット判定手段と、

前記パケット判定手段でネットワーク管理用パケットであると判断された場合、そのパケットを送信したデバイスに対して、そのデバイスが監視の対象となるデバイスであることを確認するための確認パケットを送信する送信手段と、

前記監視パケットに対する応答に基づいて、その応答を返したデバイスの状態をアドレスと共に登録するデバイスアドレス登録手段と

を実現するためのコンピュータプログラムを格納するコンピュータ可読の記憶媒体。

【発明の詳細な説明】

【０００１】

【発明の属する技術分野】

本発明は、例えばコンピュータネットワークに接続されたデバイス等を管理するためのネットワークデバイス管理装置及び方法に関する。

【０００２】

【従来の技術】

近年、コンピュータを相互に接続したローカルエリアネットワーク（ＬＡＮ）が普及しており、このようなローカルエリアネットワークは、ビルのフロアまたはビル全体、ビル群（構内）、地域、あるいはさらに大きいエリアにわたって構築することができる。このようなネットワークは更に相互接続され、世界的規模のネットワークにも接続することができる。このような相互接続されたそれぞれのＬＡＮは、多様なハードウェア相互接続技術といくつものネットワークプロトコルを持つことがある。

【０００３】

他と切り離された簡単なＬＡＮは個々のユーザが管理することができる。すなわち、ユーザが機器を取り替えたり、ソフトウェアをインストールしたり、問題点を診断したりすることができる。

【０００４】

しかし一方、規模の大きい複雑なＬＡＮや相互接続された大きなＬＡＮグループは「管理」を必要とする。「管理」とは、人間のネットワーク管理者とその管理者が使用するソフトウェアの両方による管理を意味する。本願においては、「管理」とはシステム全体を管理するためのソフトウェアによる管理を意味し、「ユーザ」とはネットワーク管理ソフトウェアを使用する人を意味するものとする。このユーザは、通常、システム管理責任者である。ユーザは、ネットワーク管理ソフトウェアを使うことによって、ネットワーク上で管理データを得て、このデータを変更することができる。

【０００５】

大規模ネットワークシステムは、通常、機器の増設と除去、ソフトウェアの更新、及び問題の検出などを絶えず行うことが必要な動的システムである。一般に、様々な人が所有する、様々な業者から供給される様々なシステムが存在する。

【０００６】

このような大規模ネットワークシステムを構成するネットワーク上のデバイスを管理するための方法として、これまでにいくつかの試みが数多くの標準機関で

なされている。国際標準化機構（ＩＳＯ）は開放型システム間相互接続（Open System Interconnection, ＯＳＩ）モデルと呼ばれる汎用基準フレームワークを提供した。ネットワーク管理プロトコルのＯＳＩモデルは、共通管理情報プロトコル（Comon Management Information Protocol, ＣＭＩＰ）と呼ばれる。ＣＭＩＰはヨーロツパの共通ネットワーク管理プロトコルである。

【０００７】

また近年では、より共通性の高いネットワーク管理プロトコルとして、簡易ネットワーク管理プロトコル（Simple Network Management Protocol, ＳＮＭＰ）と呼ばれるＣＭＩＰに関連する一変種のプロトコルがある。（「ＴＣＰ／ＩＰネットワーク管理入門実用的な管理をめざして」Ｍ．Ｔ．ローズ＝著／西田竹志＝訳（株）トッパン発行１９９２年８月２０日初版を参照。）

このＳＮＭＰネットワーク管理技術によれば、ネットワーク管理システムには、少なくとも１つのネットワーク管理ステーション（ＮＭＳ）、各々がエージェントを含むいくつかの管理対象ノード、及び管理ステーションやエージェントが管理情報を交換するために使用するネットワーク管理プロトコルが含まれる。ユーザは、ＮＭＳ上でネットワーク管理ソフトウェアを用いて管理対象ノード上のエージェントソフトウェアと通信することにより、ネットワーク上のデータを得、またデータを変更することができる。

【０００８】

ここでエージェントとは、各々のターゲット装置についてのバックラウンドプロセスとして走るソフトウェアである。ユーザがネットワーク上の装置に対して管理データを要求すると、管理ソフトウェアはオブジェクト識別情報を管理パケットまたはフレームに入れてターゲットエージェントへ送り出す。エージェントは、そのオブジェクト識別情報を解釈して、そのオブジェクト識別情報に対応するデータを取り出し、そのデータをパケットに入れてユーザに送り返す。時には、データを取り出すために対応するプロセスが呼び出される場合もある。

【０００９】

またエージェントは、自分の状態に関するデータをデータベースの形式で保持している。このデータベースのことを、ＭＩＢ(Management Information Base)

と呼ぶ。図４は、ＭＩＢの構造を示す概念図である。図４に示すように、ＭＩＢは木構造のデータ構造をしており、全てのノードが一意に番号付けされている。図４において、かっこ内に書かれている番号が、そのノードの識別子である。例えば、図４においてノード４０１の識別子は１である。ノード４０２の識別子は、ノード４０１の下の３なので、１・３と表記される。同様にして、ノード４０３の識別子は、１・３・６・１・２と表記される。このノードの識別子のことを、オブジェクト識別子(OBJECT IDENTIFIER)と呼ぶ。

【００１０】

このＭＩＢの構造は、管理情報構造（ＳＭＩ：Structure of Management Information）と呼ばれ、RFC1155 Structure and Identificatlon of Management Information for TCP/IP-based Internetsで規定されている。

【００１１】

図４には、標準として規定されているＭＩＢのうち、一部のもののみを抜き出して記載してある。

【００１２】

次に、ＳＮＭＰプロトコルについて簡単に説明する。ネットワーク管理ユーティリティソフトウェアが動作しているＰＣ（以下、マネージャと呼称する）とＳＮＭＰエージェントが動作している管理対象ネットワークデバイス（以下、エージェントと呼称する）とは、ＳＮＭＰプロトコルを用いて通信を行う。ＳＮＭＰプロトコルには５種類のコマンドがあり、それぞれＧｅｔ－ｒｅｑｕｅｓｔ、Ｇｅｔ－ｎｅｘｔ－ｒｅｑｕｅｓｔ、Ｇｅｔ－ｒｅｓｐｏｎｓｅ、Ｓｅｔ－ｒｅｑｕｅｓｔ、Ｔｒａｐと呼ばれる。これらのコマンドがマネージャとエージェントの間でやりとりされる様子を図８に示す。

【００１３】

Ｇｅｔ－ｒｅｑｕｅｓｔおよびＧｅｔ－ｎｅｘｔ－ｒｅｑｕｅｓｔは、マネージャがエージェントのＭＩＢオブジェクトの値を取得するために、マネージャがエージェントに対して送出するコマンドである。このコマンドを受け取ったエージェントは、ＭＩＢの値をマネージャに通知するために、マネージャに対してＧｅｔ－ｒｅｓｐｏｎｓｅコマンドを送出する（８０１および８０２）。

【００１４】

Ｓｅｔ－ｒｅｑｕｅｓｔは、マネージャがエージェントのＭＩＢオブジェクトの値を設定するために、マネージャがエージェントに対して送出するコマンドである。このコマンドを受け取ったエージェントは、設定結果をマネージャに通知するために、マネージャに対してＧｅｔ－ｒｅｓｐｏｎｓｅコマンドを送出する（８０３）。

【００１５】

Ｔｒａｐは、エージェントが自分自身の状態の変化をマネージャに対して通知するために、エージェントがマネージャに対して送出するコマンドである（８０４）。

【００１６】

図７に、Ｔｒａｐ以外のコマンド、即ちＧｅｔ－ｒｅｑｕｅｓｔ、Ｇｅｔ－ｎｅｘｔ－ｒｅｑｕｅｓｔ、Ｇｅｔ－ｒｅｓｐｏｎｓｅおよびＳｅｔ－ｒｅｑｕｅｓｔのフォーマットを示す。

【００１７】

７００は、ＳＮＭＰメッセージを示す。ＳＮＭＰメッセージは、バージョン７０１、コミュニティ名７０２およびＰＤＵと呼ばれる領域７０３からなる。ＰＤＵ７０３を詳細に示したものが７１０である。ＰＤＵ７１０は、ＰＤＵタイプ７１１、リクエストＩＤ７１２、エラーステータス７１３、エラーインデックス７１４およびＭＩＢ情報７１５からなる。ＰＤＵタイプ７１１には、コマンドを識別する値が格納される。即ちこのフィールドの値が０であるならばＧｅｔ－ｒｅｑｕｅｓｔ、１であるならばＧｅｔ－ｎｅｘｔ－ｒｅｑｕｅｓｔ、２であるならばＧｅｔ－ｒｅｓｐｏｎｓｅ、３であるならばＳｅｔ－ｒｅｑｕｅｓｔと識別される。また、エラーステータス７１３には、エラー情報を示す値が格納される。エラーが無い場合には、このフィールドの値は０である。また、ＭＩＢ情報７１５には、オブジェクトＩＤとその値が組みになって格納される。

【００１８】

次に、管理が必要な大規模なネットワークについて説明する。

【００１９】

図１は、プリンタをネットワークに接続するためのネットワークボード（ＮＢ）１０１を、開放型アーキテクチャを持つプリンター１０２へつなげた場合を示す図である。ＮＢ１０１はローカルエリアネットワーク（ＬＡＮ）１００へ、例えば、同軸コネクタをもつEthernetインターフェース10Base-2や、ＲＪ－４５を持つ10Base-T等のＬＡＮインターフェースを介してつながれている。

【００２０】

ＰＣ１０３やＰＣ１０４等の複数のパーソナルコンピューター（ＰＣ）もまた、１００に接続されており、ネットワークオペレーティングシステムの制御の下、これらのＰＣはＮＢ１０１と通信することができる。ＰＣの一つ、例えばＰＣ１０３を、ネットワーク管理部として使用するように指定することができる。ＰＣに、ＰＣ１０４に接続されているプリンター１０５のようなプリンターを接続してもよい。

【００２１】

また、ＬＡＮ１００にファイルサーバー１０６が接続されており、これは大容量（例えば１００億バイト）のネットワークディスク１０７に記憶されたファイルへのアクセスを管理する。プリントサーバー１０８は、接続されたプリンター１０９ａ及び１０９ｂ、又は遠隔地にあるプリンター１０５などのプリンターに印刷を行わせる。また他の図示しない周辺機器をＬＡＮ１００に接続してもよい。

【００２２】

更に詳しくは、図１に示すネットワークは、様々なネットワークメンバー間で効率良く通信を行うために、NovellやUNIXのソフトウェアなどのネットワークソフトウェアを使用することができる。どのネットワークソフトウェアを使用することも可能であるが、例えば、Novell社のNetWare（Novell社の商標。以下省略）ソフトウェアを使用することができる。このソフトウェアパッケージに関する詳細な説明は、NetWareパッケージに同梱されているオンラインドキュメンテーションを参照のこと。これは、Novell社からNetWareパッケージとともに購入可能である。

【００２３】

図１の構成について簡潔に説明すると、ファイルサーバー１０６は、ＬＡＮメンバー間でデータのファイルの受信や、記憶、キューイング、キャッシング、及び送信を行う、ファイル管理部としての役割を果たす。例えば、ＰＣ１０３及びＰＣ１０４それぞれによって作られたデータファイルは、ファイルサーバー１０６へ送られ、ファイルサーバー１０６はこれらのデータファイルを順に並べ、そしてプリントサーバー１０８からのコマンドに従って、並べられたデータファイルをプリンター１０９ａへ送信する。

【００２４】

またＰＣ１０３とＰＣ１０４はそれぞれ、データファイルの生成や、生成したデータファイルのＬＡＮ１００への送信や、また、ＬＡＮ１００からのファイルの受信や、更にそのようなファイルの表示及び／又は処理を行うことのできる、通常のＰＣで構成される。図１にパーソナルコンピューター機器が示されているが、ネットワークソフトウェアを実行するのに適切であるような、他のコンピューター機器を含んでもよい。例えば、UNIXのソフトウェアを使用している場合に、UNIXワークステーションをネットワークに含んでもよく、これらのワークステーションは、適切な状況下で、図示されているＰＣと共に使用することができる。

【００２５】

通常、ＬＡＮ１００などのＬＡＮは、一つの建物内の一つの階又は連続した複数の階でのユーザーグループ等の、幾分ローカルなユーザーグループにサービスを提供する。例えば、ユーザーが他の建物や他県に居るなど、あるユーザーが他のユーザーから離れるに従って、ワイドエリアネットワーク（ＷＡＮ）を作ってもよい。ＷＡＮは、基本的には、いくつかのＬＡＮを高速度サービス総合デジタルネットワーク（ＩＳＤＮ）電話線等の高速度デジタルラインで接続して形成された集合体である。従って、図１に示すように、ＬＡＮ１００と、ＬＡＮ１１０と、ＬＡＮ１２０とは変調／復調（ＭＯＤＥＭ）／トランスポンダー１３０及びバックボーン１４０を介して接続されＷＡＮを形成する。これらの接続は、数本のバスによる単純な電気的接続である。それぞれのＬＡＮは専用のＰＣを含み、また、必ずしも必要なわけではないが、通常はファイルサーバー及びプリントサ

ーバーを含む。

【００２６】

従って図１に示すように、ＬＡＮ１１０は、ＰＣ１１１と、ＰＣ１１２と、ファイルサーバー１１３と、ネットワークディスク１１４と、プリントサーバー１１５と、プリンター１１６及びプリンター１１７とを含む。対照的に、ＬＡＮ１２０はＰＣ１２１とＰＣ１２２のみを含む。ＬＡＮ１００と、ＬＡＮ１１０と、ＬＡＮ１２０とに接続されている機器は、ＷＡＮ接続を介して、他のＬＡＮの機器の機能にアクセスすることができる。

【００２７】

エージェントの実装例として、プリンタをネットワークに接続するためのネットワークボード上にエージェントを実装することが考えられる。これにより、プリンタをネットワーク管理ソフトウェアによる管理の対象とすることができる。ユーザは、ネットワーク管理ソフトウェアを用いて制御対象のプリンタの情報を得、また状態を変更することができる。より具体的には、例えばプリンタの液晶ディスプレイに表示されている文字列を取得したり、デフォルトの給紙カセットを変更したりすることができる。以下、エージェントを実装したネットワークボード（ＮＢ）をプリンタに接続する実施形態について説明する。図２に示すように、好ましくは、ＮＢ１０１は、プリンター１０２の内部拡張Ｉ／Ｏスロットに内蔵されており、ＮＢ１０１は、下に示す処理及びデータ記憶機能を持つ「埋め込まれた」ネットワークノードとなる。このＮＢ１０１の構成により、大きなマルチエリアＷＡＮネットワークを統括及び管理するための、特徴的な補助機能を持つという利点をもたらす。これらの補助機能は、例えば、ネットワーク上の遠隔地（ネットワーク統括者の事務所など）からのプリンター制御及び状態観察や、各印刷ジョブ後の次のユーザーのための保証初期環境を提供するためのプリンター構成の自動管理、及びプリンターの負荷量を特徴付け、あるいはトナーカートリッジの交換スケジュールを組むためにネットワークを通してアクセスできる、プリンターログ又は使用統計を含む。

【００２８】

このＮＢ設計において重要な要因は、共有メモリ等の両方向インターフェース

を介して、ＮＢ１０１からプリンター制御状態にアクセスする機能である。共有メモリ以外に、ＳＣＳＩインターフェース等のインターフェースを使用することもできる。これにより、多数の便利な補助機能のプログラムができるように、プリンター操作情報をＮＢ１０１又は外部ネットワークノードへ送出することができる。印刷画像データ及び制御情報のブロックは、ＮＢ１０１上にあるマイクロプロセッサーによって構成され、共有メモリに記述され、そして、プリンター１０２によって読み込まれる。同様に、プリンター状態情報は、プリンター１０２から共有メモリへ送られ、そこからＮＢプロセッサーによって読み込まれる。

【００２９】

図２は、ＮＢ１０１をプリンター１０２にインストールした状態を示す断面図である。図２に示すように、ＮＢ１０１はネットワーク接続の為のフェースプレート１０１ｂを設置した印刷回路ボード１０１ａから構成されており、コネクタ１７０を介してプリンターインターフェースやード１５０に接続されている。プリンターインターフェースカード１５０は、プリンター１０２のプリンターエンジンを直接制御する。印刷データ及びプリンター状態コマンドは、ＮＢ１０１からコネクタ１７０を介して、プリンターインターフェースカード１５０へ入力され、また、プリンター状態情報はプリンターインターフェースカード１５０からやはりコネクター１７０を介して得られる。ＮＢ１０１はこの情報を、フェースプレート１０１ｂのネットワークコネクタを介して、ＬＡＮ１００上で通信する。同時に、プリンター１０２は、従来のシリアルポート１０２ａ及びパラレルポート１０２ｂから、印刷データを受信することもできる。

【００３０】

図３は、ＮＢ１０１とプリンター１０２とＬＡＮ１００との電気的接続を示すブロック図である。ＮＢ１０１は、ＬＡＮ１００へはＬＡＮインターフェースを介して、プリンター１０２へはプリンターインターフェースカード１５０を介して直接接続されている。ＮＢ１０１上にはＮＢ１０１を制御するためのマイクロプロセッサー３０１と、マイクロプロセッサー３０１の動作プログラムを格納するためのＲＯＭ３０３と、マイクロプロセッサー３０１がプログラムを実行する上でワークとして用いるためのＲＡＭ３０２と、ＮＢ１０１とプリンタインタフ

ェースカード１５０とが相互にデータをやりとりするための共有メモリ２００があり、内部バスを通じて相互に接続されている。ＮＢ１０１がＳＮＭＰのエージェントとして動作するためのプログラムはＲＯＭ３０３に格納されている。マイクロプロセッサ３０１は、ＲＯＭ３０３に格納されたプログラムに従って動作し、ワークエリアとしてＲＡＭ３０２を用いる。また、プリンターインターフェースカード１５０と相互に通信するためのバッファ領域として共有メモリ２００を用いる。

【００３１】

プリンターインタフェースカード１５０上のマイクロプロセッサー１５１はＮＢ１０１とのデータのアクセスを、ＮＢ１０１に設置されている共有メモリ２００を介して行う。プリンターインタフェースカード１５０上のマイクロプロセッサー１５１は、実際に印刷機構を動かすプリンターエンジン１６０とも通信する。

【００３２】

一方、ネットワーク管理ソフトウェアが稼動するＰＣ側について、以下で説明する。

【００３３】

図５は、ネットワーク管理ソフトウェアが稼動可能なＰＣの構成を示すブロック図である。

【００３４】

図５において、５００は、ネットワーク管理ソフトウェアが稼動するＰＣであり、図１における１０３と同等である。ＰＣ５００は、ＲＯＭ５０２もしくはハードディスク（ＨＤ）５１１に記憶された、あるいはフロッピーディスクドライブ（ＦＤ）５１２より供給されるネットワーク管理プログラムを実行するＣＰＵ５０１を備え、システムバス５０４に接続される各デバイスを総括的に制御する。

【００３５】

５０３はＲＡＭで、ＣＰＵ５０１の主メモリ、ワークエリア等として機能する。

【００３６】

５０５はキーボードコントローラ（ＫＢＣ）で、キーボード（ＫＢ）５０９や不図示のポインティングデバイス等からの指示入力を制御する。５０６はＣＲＴコントローラ（ＣＲＴＣ）で、ＣＲＴディスプレイ（ＣＲＴ）５１０の表示を制御する。５０７はディスクコントローラ（ＤＫＣ）で、ブートプログラム、種々のアプリケーション、編集ファイル、ユーザファイルそしてネットワーク管理プログラム等を記憶するハードディスク（ＨＤ）５１１およびフロッピーディスクコントローラ（ＦＤ）５１２とのアクセスを制御する。５０８はネットワークインタフェースカード（ＮＩＣ）で、ＬＡＮ１００を介して、エージェントあるいはネットワーク機器と双方向にデータをやりとりする。

【００３７】

次に、従来例におけるネットワーク管理ソフトウェアの構成について説明する。

【００３８】

従来例におけるネットワーク管理装置は、図５に示したようなネットワーク管理装置を実現可能なＰＣと同様の構成のＰＣ上に実現される。ハードディスク（ＨＤ）５１１には、後述のすべての説明で動作主体となる本願に係るネットワーク管理ソフトウェアのプログラムが格納される。後述のすべての説明において、特に断りのない限り、実行の主体はハード上はＣＰＵ５０１である。一方、ソフトウェア上の制御の主体は、ハードディスク（ＨＤ）５１１に格納されたネットワーク管理ソフトウェアである。本従来例においては、ＯＳは例えば、ウィンドウズ９５（マイクロソフト社製）を想定しているが、これに限るものではない。

【００３９】

なお本願に係るネットワーク管理プログラムは、フロッピーディスクやＣＤ－ＲＯＭなどの記憶媒体に格納された形で供給されても良く、その場合には図５に示すフロッピーディスクコントローラ（ＦＤ）５１２または不図示のＣＤ－ＲＯＭドライブなどによって記憶媒体からプログラムが読み取られ、ハードディスク（ＨＤ）５１１にインストールされる。

【００４０】

図６は、本従来例に係るネットワーク管理ソフトウェアのモジュール構成図である。このネットワーク管理ソフトウェアは、図５におけるハードディスク５１１に格納されており、ＣＰＵ５０１によって実行される。その際、ＣＰＵ５０１はワークエリアとしてＲＡＭ５０３を使用する。

【００４１】

図６において、６０１はデバイスリストモジュールと呼ばれ、ネットワークに接続されたデバイスを一覧にして表示するモジュールである。（一覧表示の様子については、後ほど図１５を用いて説明する。）６０２は全体制御モジュールと呼ばれ、デバイスリストからの指示をもとに、他のモジュールを統括する。６０３はコンフィグレータと呼ばれ、エージェントのネットワーク設定に関する特別な処理を行うモジュールである。６０４は、探索モジュールと呼ばれ、ネットワークに接続されているデバイスを探索するモジュールである。探索モジュール６０４によって探索されたデバイスが、デバイスリスト６０１によって一覧表示される。６０５は、プリントジョブの状況をNetWare API６１６を用いてネットワークサーバから取得するNetWareジョブモジュールである。（なお、NetWare APIについては、例えばNovell社から発行されている"NetWare Programmer*s Guide for C"等を参照のこと。この書籍はノベル株式会社から購入可能である。）６０６および６０７は後述するデバイス詳細ウィンドウを表示するためのＵＩ(UserInterface)モジュールであり、詳細情報を表示する対象機種毎にＵＩモジュールが存在する。６０８および６０９は制御モジュールと呼ばれ、詳細情報を取得する対象機種に特有の制御を受け持つモジュールである。ＵＩモジュールと同様に、制御モジュールも詳細情報を表示する対象機種毎に存在する。制御Ａモジュール６０８および制御Ｂモジュール６０９は、ＭＩＢモジュール６１０を用いて管理対象デバイスからＭＩＢデータを取得し、必要に応じてデータの変換を行い、各々対応するＵＩＡモジュール６０６またはＵＩＢモジュール６０７にデータを渡す。

【００４２】

さて、ＭＩＢモジュール６１０は、オブジェクト識別子とオブジェクトキーとの変換を行うモジュールである。ここでオブジェクトキーとは、オブジェクト識

別子と一対一に対応する３２ビットの整数のことである。オブジェクト識別子は可変長の識別子であり、ネットワーク管理ソフトウェアを実装する上で扱いが面倒なので、本願に係るネットワーク管理ソフトウェアにおいてはオブジェクト識別子と一対一に対応する固定長の識別子を内部的に用いている。ＭＩＢモジュール６１０より上位のモジュールはこのオブジェクトキーを用いてＭＩＢの情報を扱う。これにより、ネットワーク管理ソフトウェアの実装が楽になる。

【００４３】

６１１はＳＮＭＰモジュールと呼ばれ、ＳＮＭＰパケットの送信と受信を行う。

【００４４】

６１２は共通トランスポートモジュールと呼ばれ、ＳＮＭＰデータを運搬するための下位プロトコルの差を吸収するモジュールである。実際には、動作時にユーザが選択したプロトコルによって、ＩＰＸハンドラ６１３かＵＤＰハンドラ６１４のいずれかがデータを転送する役割を担う。なお、ＵＤＰハンドラは、実装としてWinSock６１７を用いている。（WinSockについては、例えばWindows socket API v1.1の仕様書を参照のこと。このドキュメントは、複数箇所から入手可能であるが、例えばマイクロソフト社製のコンパイラであるVisual C++に同梱されている。）

コンフィグレータ６０３が用いる現在のプロトコル６１５というのは、動作時にユーザが選択しているＩＰＸプロトコルかＵＤＰプロトコルのいずれかのことを示す。

【００４５】

本従来例で使用する探索モジュール６０４とＭＩＢモジュール６１０との間のインタフェースについて説明する。

【００４６】

ＭＩＢモジュール６１０は図９に示すＣ言語のＡＰＩ(Application Program Interface)を上位モジュールに提供する。

【００４７】

最初に、上位モジュールはＭＩＢモジュールとの間で、指定したアドレスに対

するインタフェース（これをポートと呼ぶ）を開設するために、MIBOpen APＩ９０１呼び出しを行う。ＭＩＢモジュールは、開設されたインタフェースを識別するための識別子（これをポート識別子と呼ぶ）を上位モジュールに返す（MIBOpen API９０１の第一引数portに返される値）。以降上位モジュールはポート識別子を用いてＭＩＢモジュールとのやりとりを行う。

【００４８】

ここで指定するアドレスは、動作しているプロトコルのアドレスであり、ＩＰプロトコルの場合はＩＰアドレス、NetWareプロトコルの場合はNetWareアドレスである。さらにブロードキャストアドレスを指定することもできる。

【００４９】

ブロードキャストアドレスを指定してポートをオープンした場合は、ブロードキャストアドレスに応答する複数のデバイスと通信を行うことが可能である。

【００５０】

上位モジュールはポートを使用しなくなったときMIBClose API９０４呼び出しを行いポートを閉じる。

【００５１】

上位モジュールがＭＩＢオブジェクトの読み出しを行う場合は、MIBReadObjects API９０２呼び出しを行う。MIBReadObjects API９０２呼び出しにはポート識別子、読み出すべきＭＩＢオブジェクトのオブジェクトキーを指定すると共に、ＭＩＢモジュールが読み出したＭＩＢオブジェクトの値を上位へ通知するためのコールバック関数のアドレスを指定する。

【００５２】

MIBReadObjects API９０２呼び出しによりＳＮＭＰのＧｅｔ－ｒｅｑｕｅｓｔコマンドが生成され、ネットワーク上に送信される。図８に示したように、このＧｅｔ－ｒｅｑｕｅｓｔコマンドに応答するエージェントを持つデバイスはＧｅｔ－ｒｅｓｐｏｎｓｅコマンドを送信する。

【００５３】

上位モジュールがＭＩＢオブジェクトへの書き込みを行う場合は、MIBWriteObjects API９０３呼び出しを行う。MIBWriteObjects API９０３呼び出しにはポー

ト識別子、書き込むＭＩＢオブジェクトのオブジェクトキーとその値を指定すると共に、ＭＩＢモジュールが書き込みの結果を上位へ通知するためのコールバック関数のアドレスを指定する。

【００５４】

MIBWriteObjects API９０３呼び出しによりＳＮＭＰのＧｅｔ－ｒｅｑｕｅｓｔコマンドが生成され、ネットワーク上に送信される。図８に示したように、このＧｅｔ－ｒｅｑｕｅｓｔコマンドに応答するエージェントを持つデバイスはＧｅｔ－ｒｅｓｐｏｎｓｅコマンドを送信する。

【００５５】

コールバック関数はMIBReadObjects API９０２もしくはMIBWriteObjects API９０３の結果を上位モジュールに通知するためのものである。具体的には、デバイスのアドレスと受信したＧｅｔ－ｒｅｓｐｏｎｓｅコマンドの内容を上位へ通知する。

【００５６】

ブロードキャストアドレスを指定してオープンされたポートに対してMIBReadObjects API９０２を呼び出しを行った場合、ネットワーク上に送信されるＧｅｔ－ｒｅｑｕｅｓｔコマンドを運ぶパケット（ＩＰプロトコルの場合はＩＰパケット、NetWareプロトコルの場合はＩＰＸパケット）の宛先アドレスがブロードキャストアドレスになる。従って、このパケットは複数のデバイスで受信されるので、Ｇｅｔ－ｒｅｑｕｅｓｔコマンドには複数のデバイスが応答する。つまりマネージャ側では複数のＧｅｔ－ｒｅｓｐｏｎｓｅコマンドを受信する。この場合、コールバック関数は、ポート識別子は同じでデバイスのアドレスが異なる複数回の呼び出しが行われる。上位モジュールはアドレス情報を調べることにより、そのコールバックがどのデバイスからのものかを知ることができる。

【００５７】

次に具体的なデータの流れを説明する。ＭＩＢモジュール６１０では上位からの要求によりオブジェクトキーからオブジェクトＩＤへの変換等の処理を行いＳＮＭＰモジュール６１１へコマンド送信要求を行う。ＳＮＭＰモジュール６１１はＭＩＢモジュール６１０からの送信要求によりSNMP PDUをＲＡＭ５０３上で組

み立て、共通トランスポートモジュール６１２へ送信要求を行う。共通トランスポートモジュール６１２では動作プロトコルによりヘッダの付加等の所定の処理を行い、ＴＣＰ／ＩＰプロトコルであればWinSockモジュール６１７へ、NetWareプロトコルであればNetWare APIモジュール６１６へパケット送信要求を行う。以下ＴＣＰ／ＩＰプロトコルで動作しているものとして説明を行う。WinSockモジュール６１７は送信要求のあったパケットをＩＰパケット化し、ＯＳに対してネットワークへのデータ送信要求を行う。ＯＳはＲＡＭ５０３上のデータをシステムバス５０４を介してＮＩＣ５０８へ書き込む。ＮＩＣ５０８では書き込まれたデータを所定のフレーム化してＬＡＮ１００に送信する。

【００５８】

ＬＡＮ１００に接続されているデバイスからのパケットはＮＩＣ５０８で受信される。ＮＩＣ５０８ではパケット受信を割り込みによりＯＳに通知する。ＯＳはＮＩＣ５０８から受信パケットをシステムバス５０４経由で読み出しＲＡＭ５０３に置く。ＯＳでは動作プロトコルもしくは受信したパケットからプロトコルを判断し、ＴＣＰ／ＩＰプロトコルであればWinSockモジュール６１７へ、NetWareプロトコルであればNetWare APIモジュール６１６へパケット受信が通知される。以下ＴＣＰ／ＩＰプロトコルで動作しているものとして説明を行う。WinSockモジュール６１７では、受信パケットが自分宛のものかどうかを受信パケット中のアドレスにより判断する。受信パケットが自分宛のものでは無いときは受信パケットを破棄する。受信パケットが自分宛であった場合は、ＵＰＤハンドラ６１４起動し、共通トランスポートモジュール６１２にパケット受信を通知する。共通トランスポートモジュール６１２ではトランスポートヘッダの除去等の所定の処理を行い、ＳＮＭＰモジュール６１１へパケット受信を通知する。ＳＮＭＰモジュール６１１ではＳＮＭＰヘッダの除去等の所定の処理を行い、ＭＩＢモジュールへＰＤＵ受信を通知する。ＭＩＢモジュール６１０では所定処理を行うと共に受信した情報をMIB APIで規定された形式に変換し上位モジュールのコールバック関数を呼び出すことにより、デバイスからの応答を上位をモジュールへ通知する。

【００５９】

なお、以下の説明において、本願に係るネットワーク管理ソフトウェアのことを「NetSpot」と呼称する。

【００６０】

NetSpotのインストールに必要なファイルは、通常、フロッピーディスク（ＦＤ）やＣＤ－ＲＯＭなどの物理媒体に記録されて配布されるか、あるいはネットワークを経由して伝送される。ユーザは、これらの手段によりNetSpotのインストールに必要なファイルを入手した後、所定のインストール手順に従ってNetSpotのインストールを開始する。

【００６１】

NetSpotのインストール手順は、他の一般的なソフトウェアのインストール手順と同様である。すなわち、ユーザがNetSpotのインストーラをパーソナルコンピュータ（ＰＣ）上で起動すると、その後はインストーラが自動的にインストールを実行する。インストーラは、NetSpotの動作に必要なファイルをＰＣのハードディスクにコピーし、また、必要に応じてユーザから情報を入力してもらいながら、NetSpotの動作に必要なファイルの修正または新規作成なども行う。

【００６２】

次に、本従来例におけるネットワーク管理プログラムにおける探索シーケンスについて説明する。

【００６３】

図１０は、従来のネットワーク管理プログラムにおける探索シーケンスを示す図である。

【００６４】

図１０において、探索モジュール１０３０はネットワーク管理プログラムの探索モジュールを示し、これは図６における６０４と同等である。この探索モジュールは、ネットワーク管理プログラムの他のモジュールと同様に、図１におけるＰＣ１０３上で図５におけるＣＰＵ５０１によって実行される。

【００６５】

デバイス１０３１は、ネットワーク上に接続されており、かつＳＮＭＰエージェントが動作しているデバイスでプリンタとして標準的なＭＩＢのみを搭載して

いるプリンタを示し、例えば図１におけるＮＢ１１８を示す。ここで、標準的なＭＩＢとは、ＲＦＣ１２１３、ＲＦＣ１５１４およびＲＦＣ１７５９に規定されているＭＩＢなどを言う。

【００６６】

デバイス１０３２は、ネットワーク上に接続されており、かつＳＮＭＰエージェントが動作しているデバイスで、プリンタとして標準的なＭＩＢおよび企業独自のプライベートＭＩＢを搭載しているプリンタを示し、例えば図１におけるＮＢ１０１を示す。本実施例では、ＮＢ１０１が企業独自のプライベートＭＩＢを実装しているものとして説明を行なう。この企業独自のプライベートＭＩＢは、図４のノード４０８で示されるｃａｎｏｎノード以下のＭＩＢオブジェクトである。

【００６７】

上位モジュールから探索開始の指示が出されると、探索モジュールはブロードキャストアドレスを指定してデバイスの状態とデバイス種別を取得するためのＧｅｔリクエストＳＮＭＰパケットをネットワークに送出する（１００１および１００２）。このパケットは、ネットワークに接続されている全てのデバイスに届けられる。

【００６８】

ここで、問い合わせるＭＩＢオブジェクトとして、１００１では標準的なＭＩＢのみを、１００２ではプライベートＭＩＢを用いる。異なるＭＩＢオブジェクトを問い合わせるのは、プライベートＭＩＢを搭載しているデバイスと、標準ＭＩＢのみを搭載しているデバイスの両方を見つけたいためである。

【００６９】

このＳＮＭＰパケットに対して、ＳＮＭＰエージェントを実装しているネットワークデバイスは、それぞれ応答パケットを送出する（１００３から１００６）。

【００７０】

ここで１００３は、標準ＭＩＢを問い合わせるブロードキャストパケット１００１に対するデバイス１０３１の応答であり、デバイス１０３１の状態に応じた

値が返る。１００４は、標準ＭＩＢを問い合わせるブロードキャストパケット１００１に対するデバイス１０３２の応答であり、デバイス１０３２の状態に応じた値が返る。１００５は、プライベートＭＩＢを問い合わせるブロードキャストパケット１００２に対するデバイス１０３１の応答であり、デバイス１０３１はプライベートＭＩＢを実装していないため、エラーとしてnoSuchNameが返される。１００６は、プライベートＭＩＢを問い合わせるブロードキャストパケット１００２に対するデバイス１０３２の応答であり、デバイス１０３２の状態に応じた値が返る。

【００７１】

探索を開始してから一定の時間が経過すると一時応答タイマー１０２３が満了するので、探索モジュールはさらに詳細な情報を取得すべく、それぞれのデバイスに対してＳＮＭＰパケットを送信する。

【００７２】

より具体的には、一時探索タイマー１０２３が満了した時点までに、デバイス１０３１からはプライベートＭＩＢに対してnoSuchNameエラーが返ってきたことがわかっているので、このデバイスをプライベートＭＩＢには対応していないデバイス、すなわち標準ＭＩＢのみに対応しているデバイスとみなして、標準ＭＩＢを用いてより詳細な取得を行なう。この取得およびデバイス１０３１の返答が１００７および１００８である。

【００７３】

同様に、デバイス１０３２からは標準ＭＩＢおよびプライベートＭＩＢの両方から正常な応答が返ってきているので、このデバイスをプライベートＭＩＢを実装しているデバイスとみなして、プライベートＭＩＢを用いてより詳細な取得を行なう。

【００７４】

この取得およびデバイス１０３２の返答が１００９および１０１０である。さらに時間が経過してデバイス応答タイマー１０２１が満了した時点で、上位モジュールに対してそれまでに探索したデバイスの情報を通知する。

【００７５】

さらに時間が経過し、探索間隔タイマー１０２２が満了した時点で、再度探索を開始する。以降今まで説明した動作と全く同一のシーケンスであるので、説明を省略する。

【００７６】

【発明が解決しようとする課題】

しかしながら上記従来例では、デバイス管理装置が能動的にブロードキャストパケットを送信することによりデバイスの探索を行なう。

【００７７】

このようにブロードキャストパケットを送信することは、そのパケットに対してＳＮＭＰに応答できる全てのネットワーク機器が返信パケットを送信することになり、ネットワークのトラフィックを著しく大きくしてしまう。定期的にブロードキャストパケットを送信することはブロードキャストパケットによるトラフィック増大現象が定期的に起きることを意味し、そのような状況はネットワークの負荷および管理の観点から好ましくない。ブロードキャストパケットの送信は、できる限り少ないことが望ましい。

【００７８】

本発明は上記従来例に鑑みてなされたもので、ネットワークの負荷を増大させずにネットワークデバイスの情報を取得することがでできるネットワークデバイス管理装置及び方法を提供することを目的とする。

【００７９】

【課題を解決するための手段】

上記目的を達成するために、本発明は次のような構成からなる。すなわち、

起動後に少なくとも１度はネットワーク管理用パケットをブロードキャストするデバイスが接続されたネットワークの管理装置であって、

パケットを受信するパケット受信手段と、

前記パケット受信手段で受信したパケットが、ネットワーク管理用のパケットであるか判定するパケット判定手段と、

前記パケット判定手段でネットワーク管理用パケットであると判断された場合、そのパケットから、そのパケットを送信したデバイスのアドレスを取得するデ

バイスアドレス取得手段と、

前記デバイスアドレスを登録するデバイスアドレス登録手段とを備える。

【００８０】

あるいは、起動後に少なくとも１度はネットワーク管理用パケットをブロードキャストするデバイスが接続されたネットワークの管理装置であって、

パケットを受信するパケット受信手段と、

前記パケット受信手段で受信したパケットが、ネットワーク管理用のパケットであるか判定するパケット判定手段と、

前記パケット判定手段でネットワーク管理用パケットであると判断された場合、そのパケットを送信したデバイスに対して、そのデバイスが監視の対象となるデバイスであることを確認するための確認パケットを送信する送信手段と、

前記監視パケットに対する応答に基づいて、その応答を返したデバイスの状態をアドレスと共に登録するデバイスアドレス登録手段とを備える。

【００８１】

また好ましくは、前記応答は、その応答元のデバイスが管理対象のデバイスであれば正常応答であり、該デバイスの稼働状態を示す情報を含む。

【００８２】

また好ましくは、前記送信手段は、プリンタの状態を取得するための確認パケットを送信し、前記アドレス登録手段は、確認パケットに対する応答が正常応答であれば、応答元のデバイスのアドレスと共に、そのデバイスがプリンタであることを登録する。

【００８３】

【発明の実施の形態】

（第１の実施形態）

従来例の説明で述べたように、ＳＮＭＰの規定には、トラップと呼ばれるパケット構造がある。トラップにはいくつかの種類があるが、それらのうちコールドスタートトラップと呼ばれるものは、機器が起動した時にネットワーク上にブロードキャストされる。本発明では、このコールドスタートトラップを捉えることにより、ネットワークに新しいデバイスが追加されたことを検知しようとするも

のである。

【００８４】

以下、図面を用いて詳細に説明する。

【００８５】

図１１は、本実施例において、各種の機器のネットワーク上での配置を示す図である。

【００８６】

１１０１は、ネットワークデバイス管理プログラムが実行されるコンピュータ、１１０２はディレクトリサービスが動作しているディレクトリサーバ、１１０３はトラップパケットを監視するトラップ監視プログラムを実行するトラップ監視プログラム実行マシンである。１１０１、１１０２および１１０３のコンピュータの構成は、従来例の説明において図５で示したものと同様の構成が使える。ただし、ディレクトリサーバ１１０２およびトラップ監視プロセス実行マシン１１０３においては、キーボードコントローラ５０５、キーボード５０９、ＣＲＴコントローラ５０６、ＣＲＴ５１０のように、ユーザとの対話を行なうための入出力関連の部分は必ずしも必要ではない。さらに、１１０１、１１０２および１１０３は、同一のコンピュータであっても良い。

【００８７】

なお、上記で言うディレクトリサービスとは、言わばネットワークに関する電話帳であり、様々な情報を格納するためのものである。ディレクトリシステムの具体例としては、例えばＬＤＡＰ(Lightweight Directory Access Protocol)がある。ＬＤＡＰの規定は、ＩＥＴＦが発行しているＲＦＣ１７７７に記載されている。また解説書としては、例えば株式会社プレンティスホールより「ＬＤＡＰ　インターネット　ディレクトリ　アプリケーション　プログラミング」が１９９７年１１月１日に発行されている。

【００８８】

本実施例では、ディレクトリサーバとしてＬＤＡＰサーバを用いることにして説明を行なうが、ＬＤＡＰサーバ以外のディレクトリサーバでも本発明の本質を損なわない。

【００８９】

図１２は、トラップ監視プログラム実行マシン１１０３で動作する、トラップの監視プログラムの動作を示すフローチャートである。トラップ監視プログラムは、トラップ監視プログラム実行マシン１１０３が起動すると同時に起動するように設定されているか、あるいはネットワークの管理者によって明示的に起動されるものとする。

【００９０】

トラップ監視プログラムでは、まずステップＳ１２０１でディレクトリサーバ１１０２にネットワークを介して接続し、情報を登録する準備をする。接続するディレクトリサーバのアドレスは、システム管理者によってあらかじめ登録されており、例えばファイルに記録されてハードディスク５１１に格納されているものとする。次にステップＳ１２０２で、パケットの受信があったかどうかを調べる。これは、ＯＳが提供するＡＰＩを呼び出すことにより、ＯＳがＮＩＣ５０８の状態を調べることによって行われる。パケットの受信があった場合ステップＳ１２０３に進み、受信パケットがＳＮＭＰトラップであるかどうかを判断する。受信パケットのフォーマットが図７で説明したＳＮＭＰパケットに合致しており、かつＰＤＵタイプ７１１のフィールドの値がトラップを示す値であったときにＳＮＭＰトラップパケットであると判断する。ステップＳ１２０３で受信パケットがＳＮＭＰトラップであると判断された場合にはステップＳ１２０４に進む。ステップＳ１２０４では、受信パケットの送信元アドレスフィールドを見て、送信したデバイスのアドレスを取り出す。次にステップＳ１２０５に進み、ステップＳ１２０４で取り出したデバイスのアドレスをディレクトリサーバ１１０２に送信してディレクトリサービスに登録する。

【００９１】

一方、ステップＳ１２０３でＳＮＭＰトラップ受信ではないと判断された場合、ステップＳ１２０６に進みＳＮＭＰトラップ受信以外の受信処理を行なう。あるいは、ステップＳ１２０２でパケット受信ではないと判断された場合には、ステップＳ１２０７に進み、パケット受信以外のその他の処理を行なう。

【００９２】

ステップＳ１２０５、Ｓ１２０６あるいはＳ１２０７の処理が終了したら、ステップＳ１２０２に戻って処理を続ける。

【００９３】

図１３は、クライアント１１０１で実行されるデバイス制御プログラムにおいて、デバイスリストを表示する動作を示すフローチャートである。図１３の手順は、クライアント１１０１においてデバイス管理プログラムが起動した場合、あるいはユーザの操作によって明示的に起動される。

【００９４】

デバイスリスト表示動作においては、まずステップＳ１３０１で、ディレクトリサーバに接続する。接続するディレクトリサーバのアドレスは、システム管理者によってあらかじめ登録されており、例えばファイルに記録されてハードディスク５１１に格納されているものとする。そのアドレスは、トラップ監視プログラム実行マシン１１０３のトラップ監視プログラムが接続するディレクトリサーバのアドレスと同じである。次にステップＳ１３０２において、ステップＳ１３０１で接続したディレクトリサーバから、ディレクトリサービスに登録されている情報を取り出す。次にステップＳ１３０３で、ＣＲＴＣ５０６を操作してＣＲＴ５１０上にステップＳ１３０２で取得した情報を表示する。

【００９５】

図１４は、ステップＳ１３０３における表示の様子を示す図である。ウィンドウ１４０１に、ディレクトリサーバから取得したデバイスのアドレス一覧が表示されている。

【００９６】

以上のようにして、トラップ監視プロセスにより、ネットワーク上のデバイスが起動時にブロードキャストするコールドスタートトラップを補足してそのデバイスのアドレスを取得し、それをディレクトリサーバに登録する。こうすることで、デバイスの探索を能動的に行うことなくネットワークに接続されているデバイスのアドレスを把握できる。このため、能動的に探索パケットの発行が不要となり、ネットワークのトラフィックが軽減できる。

【００９７】

（第２の実施の形態）

トラップパケットは、ＳＮＭＰによるデバイス管理において一般的なものであるので、トラップパケットが、探索の対象としているデバイス種別からのものであることを確認してからディレクトリサービスに登録した方が、後にクライアントがディレクトリサービスにアクセスすることを考えると都合がよい。つまり、例えばクライアントがネットワークプリンタのみを制御対象にしているような場合には、トラップパケットを送信したデバイスがプリンタであることを確認してからディレクトリサービスに登録する方がより望ましい。

【００９８】

さらにトラップパケットを受信した時には、デバイスの種別を確認するためのパケットとしてデバイスの状態を問い合わせるパケットを用いることにより、デバイスに関するより詳細な情報を得ることができる。

【００９９】

本発明第２の実施例は、これらの点の改良を図ったものである。

【０１００】

以下第２の実施例においては、制御対象のネットワークデバイスがネットワークプリンタであるとして説明を進める。

【０１０１】

図１５は、本発明の第２の実施例において、トラップ監視プログラム実行マシン１１０３で実行されるトラップ監視プログラムの動作について説明したフローチャートである。

【０１０２】

この図においてステップＳ１５０１からＳ１５０４までは、第１の実施例において図１２を用いて説明したＳ１２０１からＳ１２０４までの動作と全く同じであるので説明を省略する。同様に、Ｓ１５０８とＳ１２０６およびＳ１５０９とＳ１２０７はそれぞれ同じであるので説明を省略する。

【０１０３】

ステップＳ１５０５においては、トラップパケットを送信してきたデバイスがネットワークプリンタであることを確認するために、ネットワークプリンタがイ

ンプリメントしているであろうＭＩＢオブジェクトを取得するためのＳＮＭＰパケットを送信する。より具体的には、例えばRFC1514 Host Resources MIBに規定されているhrPrinterDetectedErrorStateを取得すべくＧｅｔ－Ｎｅｘｔパケットを送信する。このＭＩＢオブジェクトは、プリンタの状態を表すものであり、デバイスに問い合わせる情報としてこのＭＩＢオブジェクトを用いることにより、プリンタであることを確認するとともに、同時にデバイスに関するより詳細な情報を得ることができる。

【０１０４】

ステップＳ１５０３でＳＮＭＰトラップ以外のパケットの受信であると判断されたら、ステップＳ１５０６に進み、そのパケットがＳＮＭＰパケットで、かつネットワークプリンタを示す正常応答であるかどうかを判断する。より具体的には、受信したパケットがステップＳ１５０５で送信したＧｅｔ－Ｎｅｘｔパケットに対するＧｅｔ－Ｒｅｓｐｏｎｓｅ応答パケットであることをリクエストＩＤフィールド７１２の値が同じであることを確認することにより行なうとともに、ＭＩＢ情報フィールド７１５に含まれている取得したＭＩＢのオブジェクトＩＤがhrPrinterDetectedErrorState.Xであることを確認する。ここでXは、トラップパケットを送信したデバイスのＭＩＢエージェントにおけるプリンタデバイスを示すインデックス値を示す。

【０１０５】

もし取得したＭＩＢのオブジェクトＩＤがhrPrinterDetectedErrorState.Xであった場合には、トラップパケットを送信したデバイスをネットワークプリンタであると判断し、ステップＳ１５０７に進む。一方、オブジェクトＩＤがhrPrinterDetectedErrorStateの次のオブジェクトであった場合、あるいはエラーステータスフィールド７１３の値がオブジェクトなし(noSuchObject)であった場合には、トラップパケットを送信したデバイスはプリンタではないと判断し、ステップＳ１５０８に進む。

【０１０６】

ステップＳ１５０７では、ステップＳ１５０１で接続したＬＤＡＰサーバに、トラップパケットを送信したデバイスのネットワークアドレス、そのデバイスが

プリンタであることを示すフラグおよびステップＳ１５０６で取得したhrPrinterDetectedErrorState.Xの値を登録する。

【０１０７】

図１６は、ステップＳ１５０７で登録した情報をクライアント１１０１が表示している様子を示した図である。表示手順については、図１３で説明したのと同様であるので省略する。

【０１０８】

図１３の情報と比較して、デバイスがプリンタであることがアイコンで示されている他、デバイスの状態に関する情報が追加されている。例えば１６０３は、デバイスに何らかの警告エラーが生じていることを示している。１６０２は、デバイスタイプがプリンタであるとの確認がとれていないデバイスを示している。

【０１０９】

以上のようにして、トラップ監視プロセスにより、ネットワーク上のデバイスが起動時にブロードキャストするコールドスタートトラップを補足してそのデバイスのアドレスを取得し、そのアドレスを有するデバイスに対して、その状態を問い合わせる。この問合せに対する応答に応じて、アドレスと共に得られた状態をディレクトリサーバに登録する。こうすることで、デバイスの探索を能動的に行うことなくネットワークに接続されているデバイスのアドレスを把握できる。このため、能動的に探索パケットの発行が不要となり、ネットワークのトラフィックが軽減できる。それに加えて、状態の問合せにより、そのデバイスの種類や稼働状態をディレクトリに登録することができ、利用者はより詳細な情報を獲得することができる。

【０１１０】

上記で説明した本発明に係るネットワークデバイス探索プログラムは、外部からインストールされるプログラムによって、ＰＣ５００によって遂行されても良い。その場合、そのプログラムはＣＤ－ＲＯＭやフラッシュメモリやフロッピーディスクなどの記憶媒体により、あるいは電子メールやパソコン通信などのネットワークを介して、外部の記憶媒体からプログラムを含む情報群をＰＣ５００上にロードすることにより、ＰＣ５００に供給される場合でも本発明は適用される

ものである。

【０１１１】

図１７は、記憶媒体の一例であるＣＤ－ＲＯＭのメモリマップを示す図である。９９９９はディレクトリ情報を記憶してある領域で、以降のインストールプログラムを記憶してある領域９９９８およびネットワークデバイス探索プログラムを記憶してある領域９９９７の位置を示している。９９９８は、インストールプログラムを記憶してある領域である。９９９７は、ネットワークデバイス制御プログラムを記憶してある領域である。本発明のネットワーク探索プログラムがＰＣ５００にインストールされる際には、まずインストールプログラムを記憶してある領域９９９８に記憶されているインストールプログラムがシステムにロードされ、ＣＰＵ５０１によって実行される。次に、ＣＰＵ５０１によって実行されるインストールプログラムが、ネットワークデバイス探索プログラムを記憶してある領域９９９７からネットワークデバイス探索プログラムを読み出して、ハードディスク５１１に格納する。

【０１１２】

なお、本発明は、複数の機器（例えばホストコンピュータ、インタフェース機器、リーダなど）から構成されるシステムあるいは統合装置に適用しても、ひとつの機器からなる装置に適用してもよい。

【０１１３】

また、前述した実施形態の機能を実現する、図１２、図１３あるいは図１５に記載された手順のプログラムコードを記録した記憶媒体を、システムあるいは装置に供給し、そのシステムあるいは装置のコンピュータ（またはＣＰＵやＭＰＵ）が記憶媒体に格納されたプログラムコードを読み出し実行することによっても、本発明の目的が達成される。

【０１１４】

この場合、記憶媒体から読み出されたプログラムコード自体が本発明の新規な機能を実現することになり、そのプログラムコードを記憶した記憶媒体は本発明を構成することになる。

【０１１５】

プログラムコードを供給するための記憶媒体としては、例えば、フロッピーディスク、ハードディスク、光ディスク、光磁気ディスク、ＣＤ－ＲＯＭ、ＣＤ－Ｒ、磁気テープ、不揮発性のメモリカード、ＲＯＭなどを用いることができる。

【０１１６】

また、コンピュータが議み出したプログラムコードを実行することによって、前述した実施形態の機能が実現される他、そのプログラムコードの指示に基づき、コンピュータ上で稼動しているＯＳなどが実際の処理の一部または全部を行い、その処理によっても前述した実施形態の機能が実現され得る。

【０１１７】

さらに、記憶媒体から読み出されたプログラムコードが、コンピュータに挿入された機能拡張ボードやコンピュータに接続された機能拡張ユニットに備わるメモリに書き込まれた後、そのプログラムコードの指示に基づき、その機能拡張ボードや機能拡張ユニットに備わるＣＰＵなどが実際の処理の一部または全部を行い、その処理によっても前述した実施形態の機能が実現され得る。

【０１１８】

なお、本発明は、前述した実施形態の機能を実現するソフトウェアのプログラムコードを記録した記憶媒体から、そのプログラムをパソコン通信など通信ラインを介して要求者にそのプログラムを配信する場合にも適用できる。

【０１１９】

【発明の効果】

以上説明したように、本発明によれば、デバイス管理装置が能動的にブロードキャストパケットを送信することなくネットワークデバイスの情報を取得することができる。このため、デバイス探索のためにネットワークの負荷が増大することを防止できる。

【図面の簡単な説明】

【図１】

プリンタをネットワークに接続するためのネットワークボードを、開放型アーキテクチャを持つプリンターへつなげた場合を示す図である。

【図２】

エージェントを実装したネットワークボードをプリンタに接続する実施形態を示す断面図である。

【図３】

ネットワークボードとプリンターとＬＡＮとの電気的接続を示すブロック図である。

【図４】

ＭＩＢの構造を示す概念図である。

【図５】

ネットワーク管理ソフトウェアが稼動可能なＰＣの構成を示すブロック図である。

【図６】

ネットワーク管理ソフトウェアのモジュール構成図である。

【図７】

ＳＮＭＰメッセージのフォーマットである。

【図８】

マネージャとエージェント間でのＳＮＭＰコマンドのやりとりを示す図である。

【図９】

ＭＩＢモジュールＡＰＩを示す図である。

【図１０】

従来例における探索モジュールとデバイス間の通信シーケンスを示す図である。

【図１１】

機器のネットワーク上での配置を示す図である。

【図１２】

トラップの監視プログラムの動作を示すフローチャートである。

【図１３】

デバイスリストを表示する動作を示すフローチャートである。

【図１４】

デバイス一覧の表示の様子を示す図である。

【図１５】

第２の実施例におけるトラップの監視プログラムの動作を示すフローチャートである。

【図１６】

第２の実施例におけるデバイス一覧の表示の様子を示す図である。

【図１７】

本発明のネットワーク管理ソフトウェアの記憶媒体におけるメモリマップを示す図である。

【符号の説明】

Ｓ１１０１　クライアント

Ｓ１１０２　ディレクトリサーバ

Ｓ１１０３　トラップ監視プログラム実行マシン

Ｓ１１０４　ネットワークデバイス

Ｓ１２０２　パケット受信手段

Ｓ１２０３　トラップパケット受信判断手段

Ｓ１２０４　デバイスアドレス取得手段

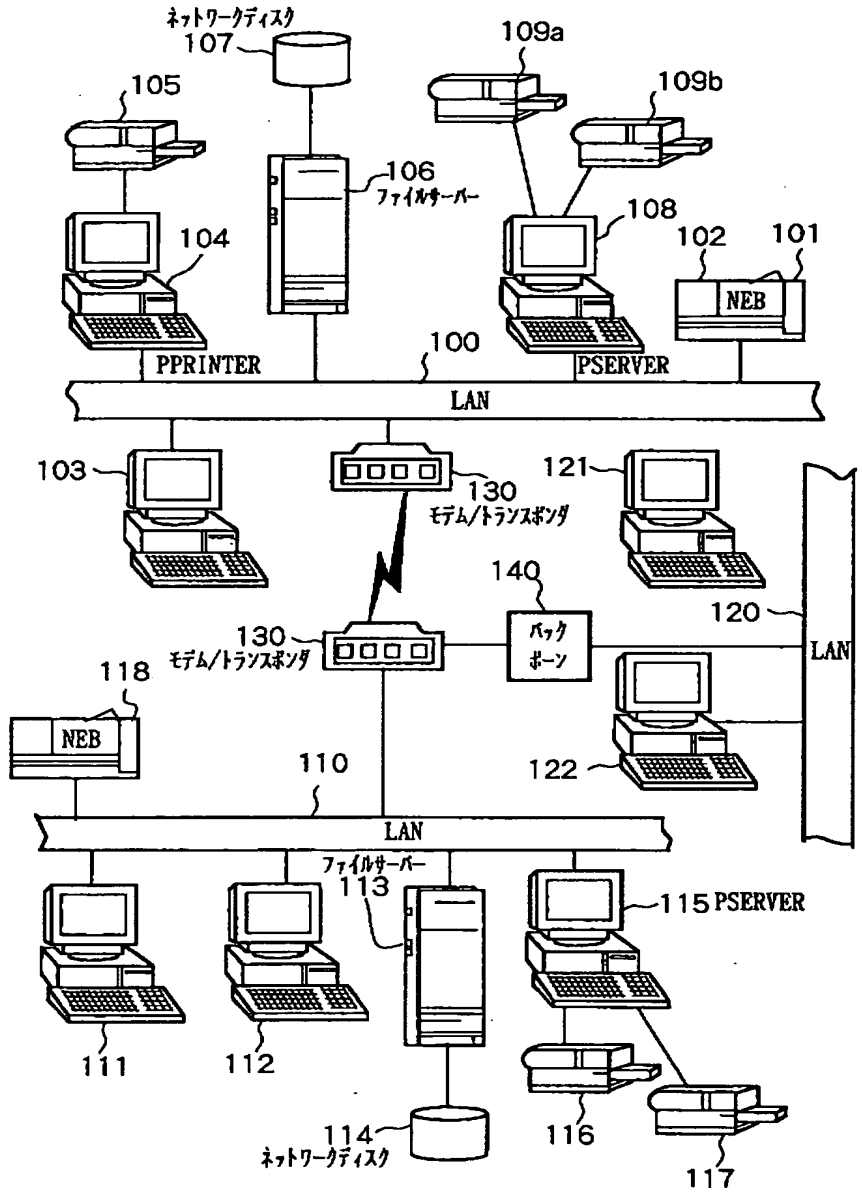
Ｓ１２０５　デバイスアドレス登録手段

Ｓ１５０５　確認パケット送信手段

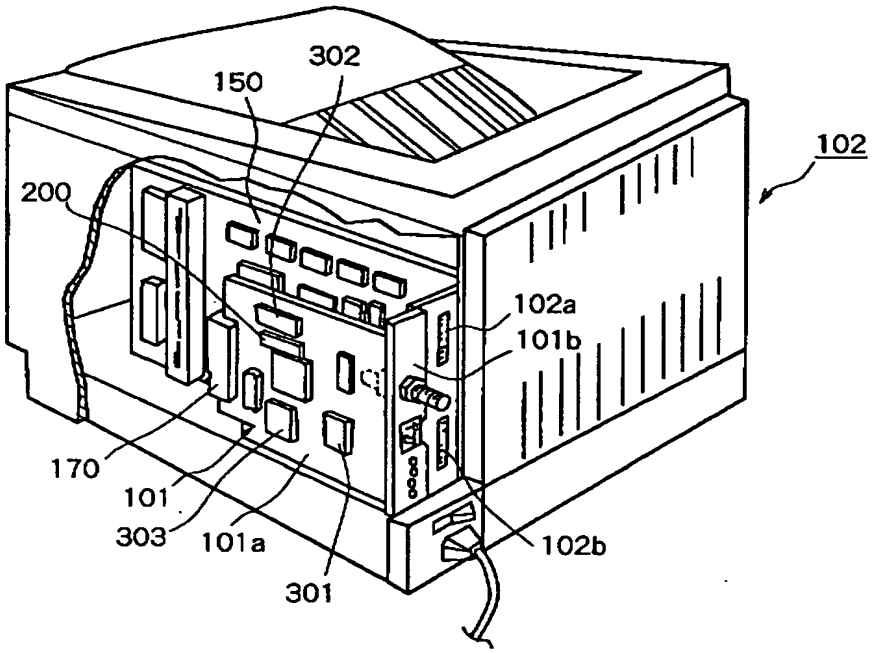
Ｓ１５０６　応答パケット確認手段

【書類名】　図面

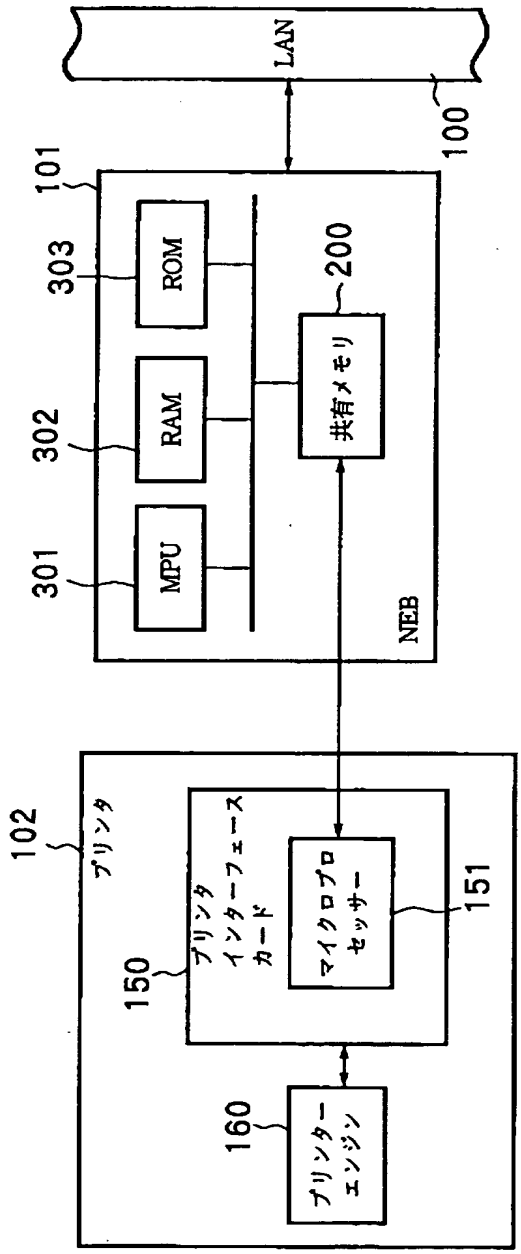
【図１】


ネットワークディスク
107
109a
109b
105
106
ファイルサーバー
108
104
102
101
NEB
PPRINTER
100
PSERVER
LAN
103
130
モデム/トランスポンダ
121
140
120
LAN
130
モデム/トランスポンダ
バックボーン
118
NEB
110
122
LAN
ファイルサーバー
113
115 PSERVER
111
112
114
ネットワークディスク
116
117

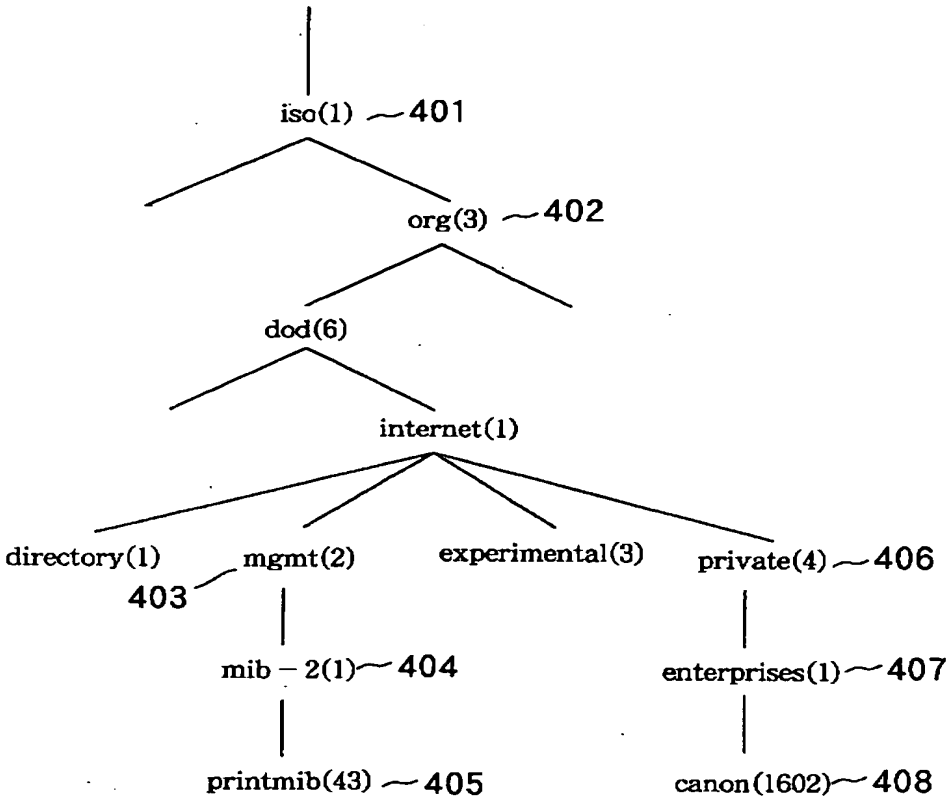
【図２】



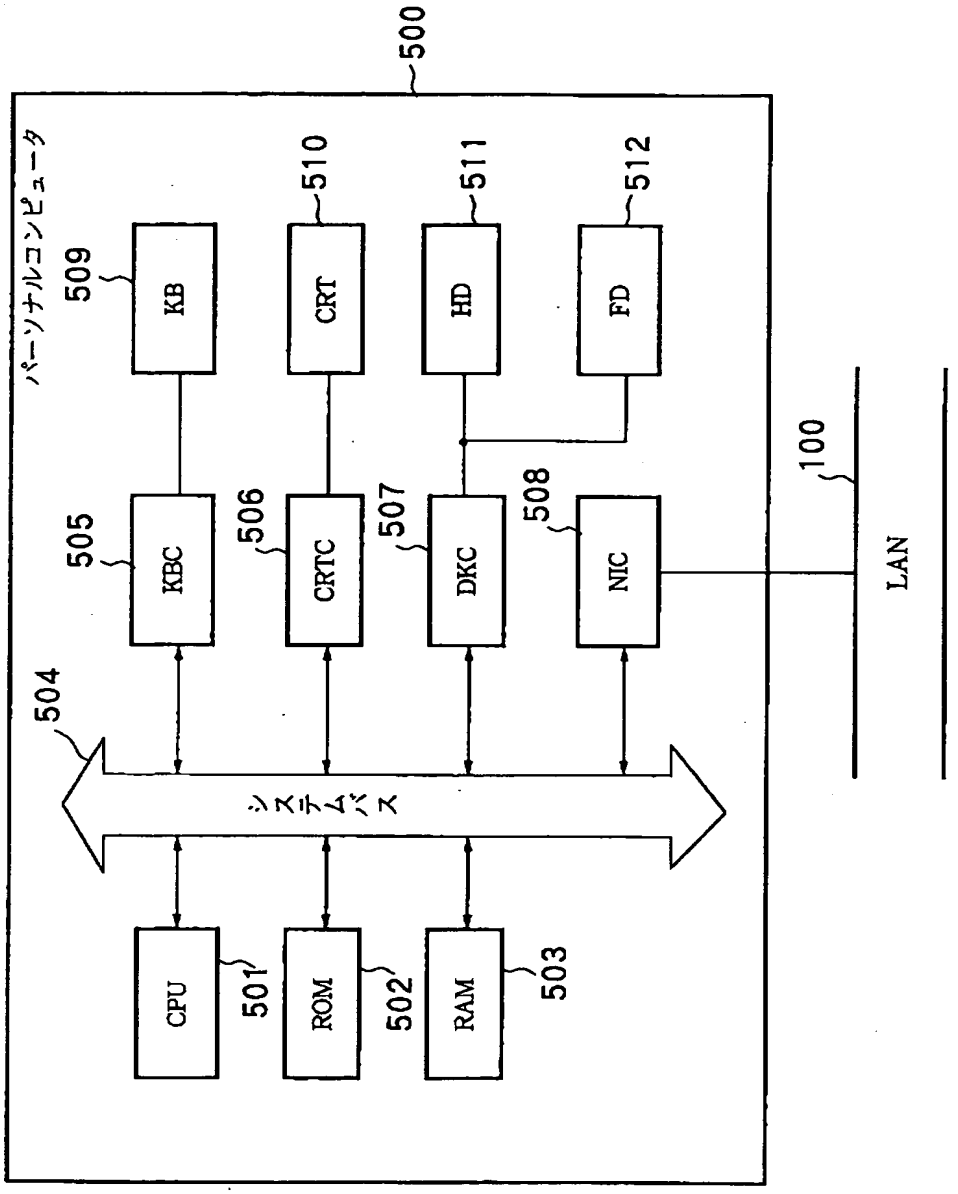
【図３】


LAN
100
101
303
ROM
200
共有メモリ
302
RAM
301
MPU
NEB
102
プリンタ
プリンタ
インターフェース
カード
150
マイクロプロ
セッサー
151
160
プリンター
エンジン

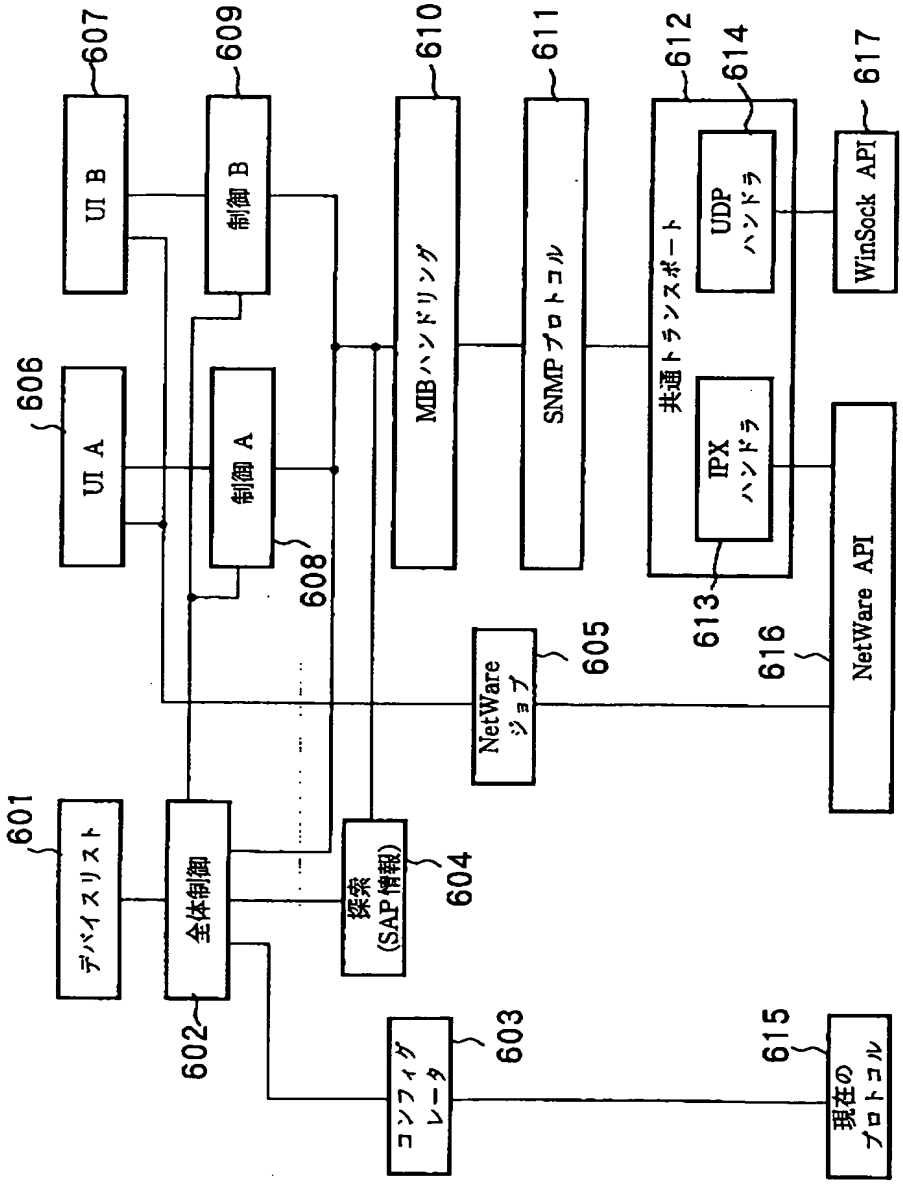
【図４】


iso(1) ～401
org(3) ～402
dod(6)
internet(1)
directory(1)
mgmt(2)
403
experimental(3)
private(4) ～406
mib－2(1)～404
enterprises(1)～407
printmib(43) ～405
canon(1602)～408

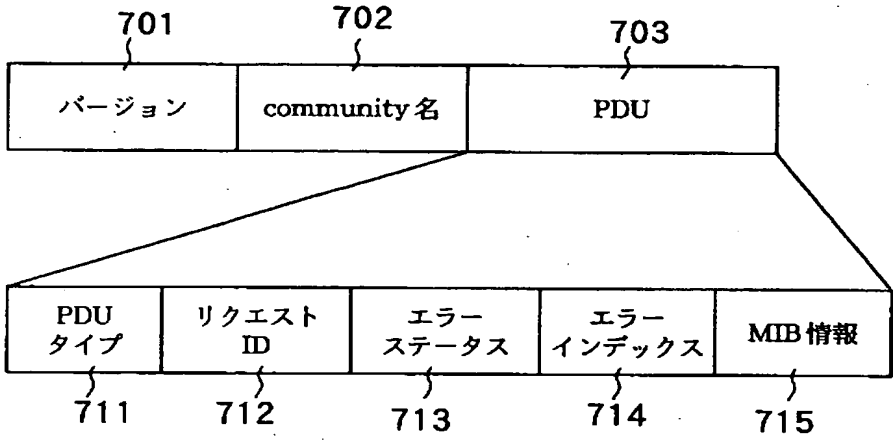
【図５】


500
パーソナルコンピュータ
509
KB
510
CRT
511
HD
512
FD
505
KBC
506
CRTC
507
DKC
508
NIC
100
LAN
504
システムバス
CPU
501
ROM
502
RAM
503

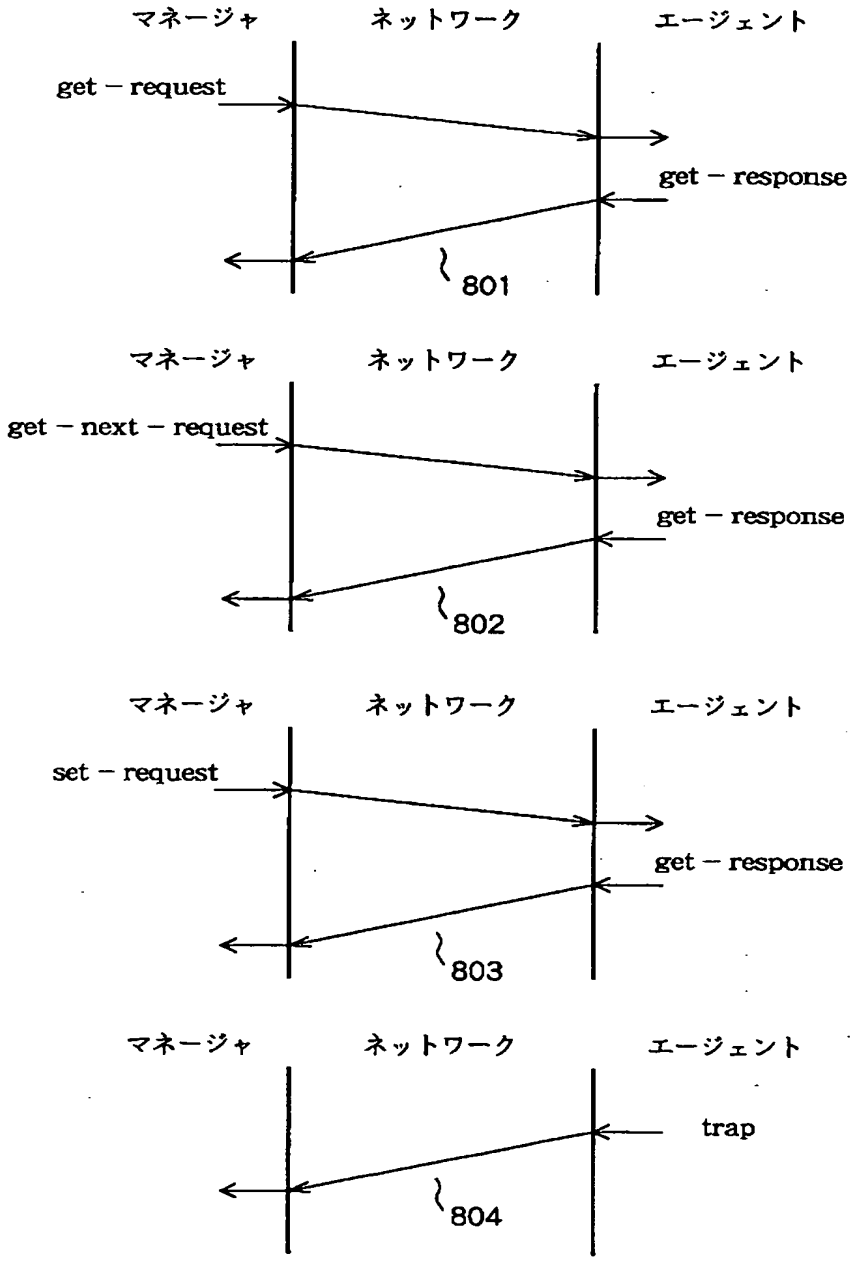
【図６】


607
609
610
611
612
614
617
UI B
制御 B
MIBハンドリング
SNMPプロトコル
共通トランスポート
UDP ハンドラ
WinSock API
606
UI A
制御 A
608
IPX ハンドラ
613
NetWare API
605
616
NetWare ジョブ
601
デバイスリスト
全体制御
探索 (SAP情報)
604
602
603
615
コンフィグ レータ
現在の プロトコル

【図７】


701
702
703
バージョン
community名
PDU
PDU
タイプ
リクエスト
ID
エラー
ステータス
エラー
インデックス
MIB情報
711
712
713
714
715

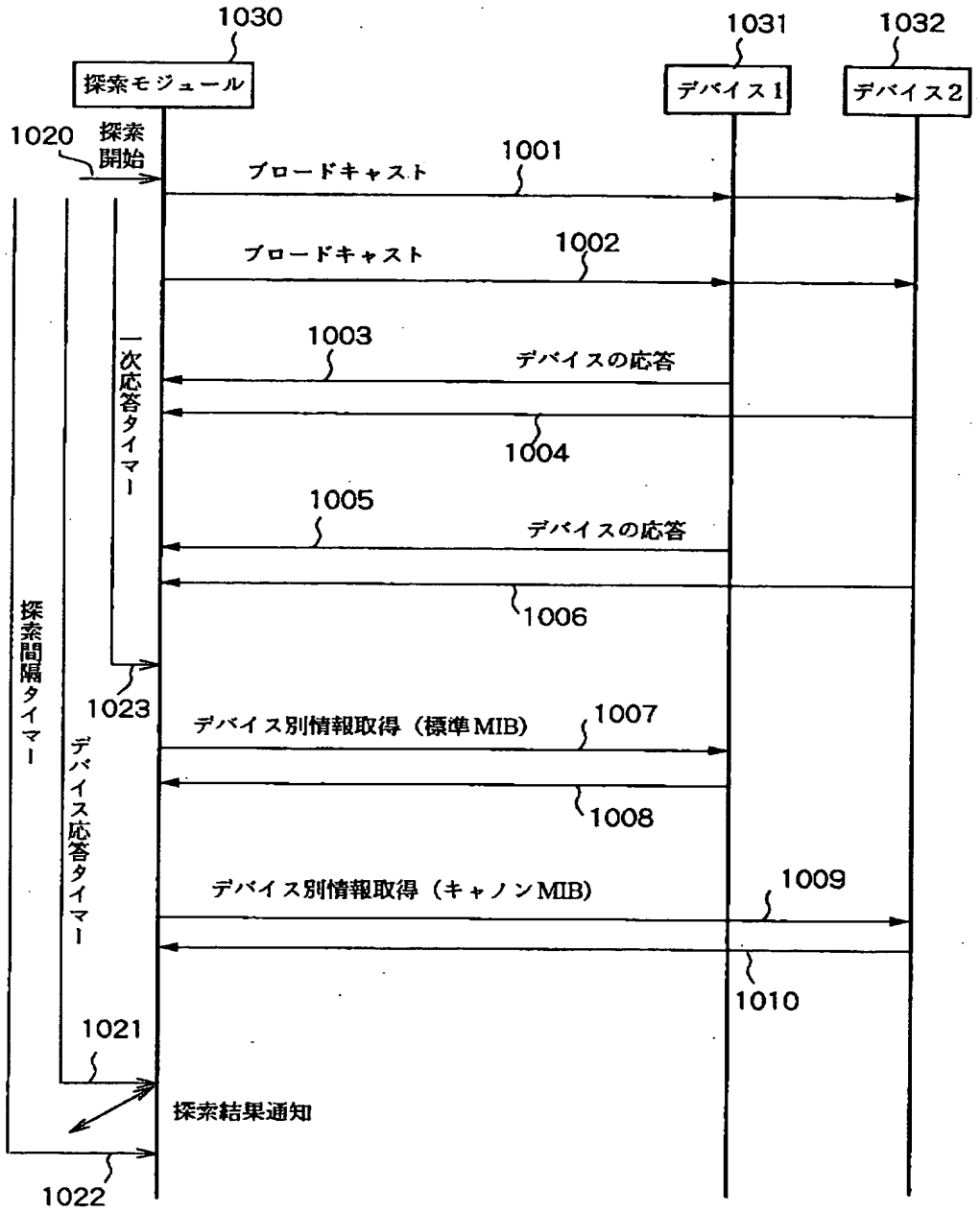
【図８】


マネージャ
ネットワーク
エージェント
get－request
get－response
801
マネージャ
ネットワーク
エージェント
get－next－request
get－response
802
マネージャ
ネットワーク
エージェント
set－request
get－response
803
マネージャ
ネットワーク
エージェント
trap
804

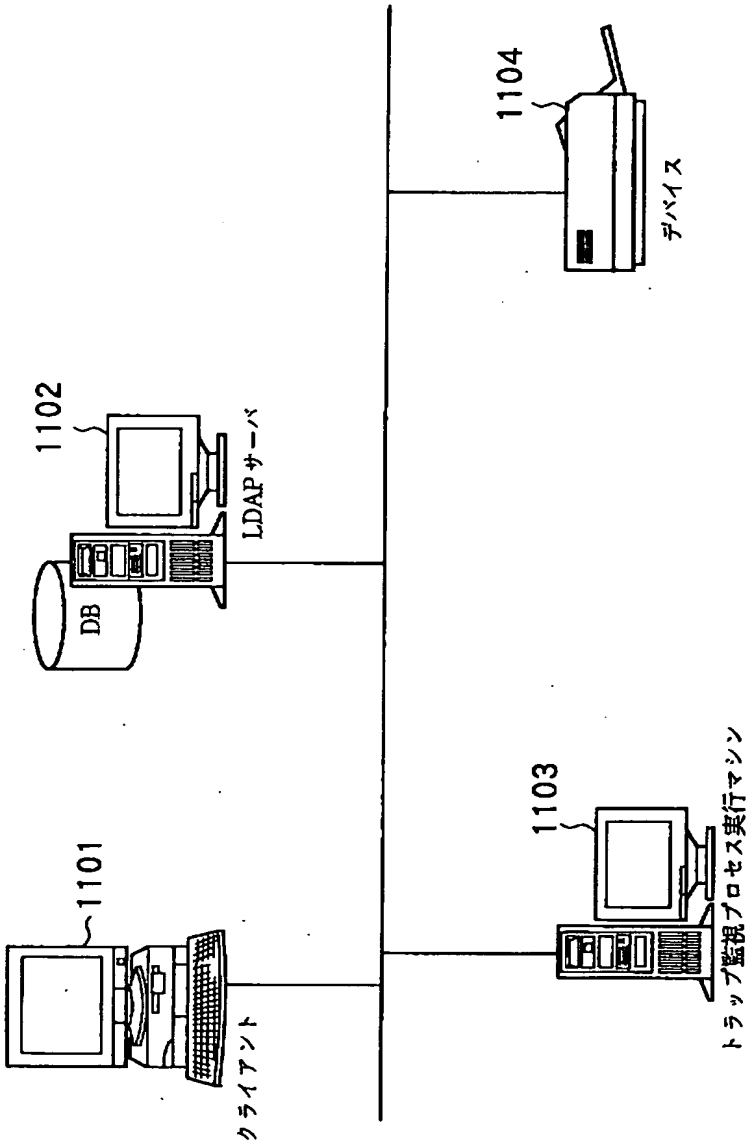
【図９】

```
                                                    901
BOOL MIBOpen (int *port, ADDR  ADDR *addr ) ;
                                                        902
BOOL MIBReadObjects (int  port,  int count,  MIBOBJ *obj,
                          CALLBACK respproc ) ;
                                                    903
BOOL MIBWriteObjects (int  port,  int count,  MIBOBJVAL *objval,
                        CALLBACK respproc ) ;
                          904
BOOL MIBClose( int port ) ;
                                                          905
typedef VOID (*CALLBACK) (int  port,  ADDR *addr,  INT result,
                              int count,  MIBOBJVAL *objval ) ;
```

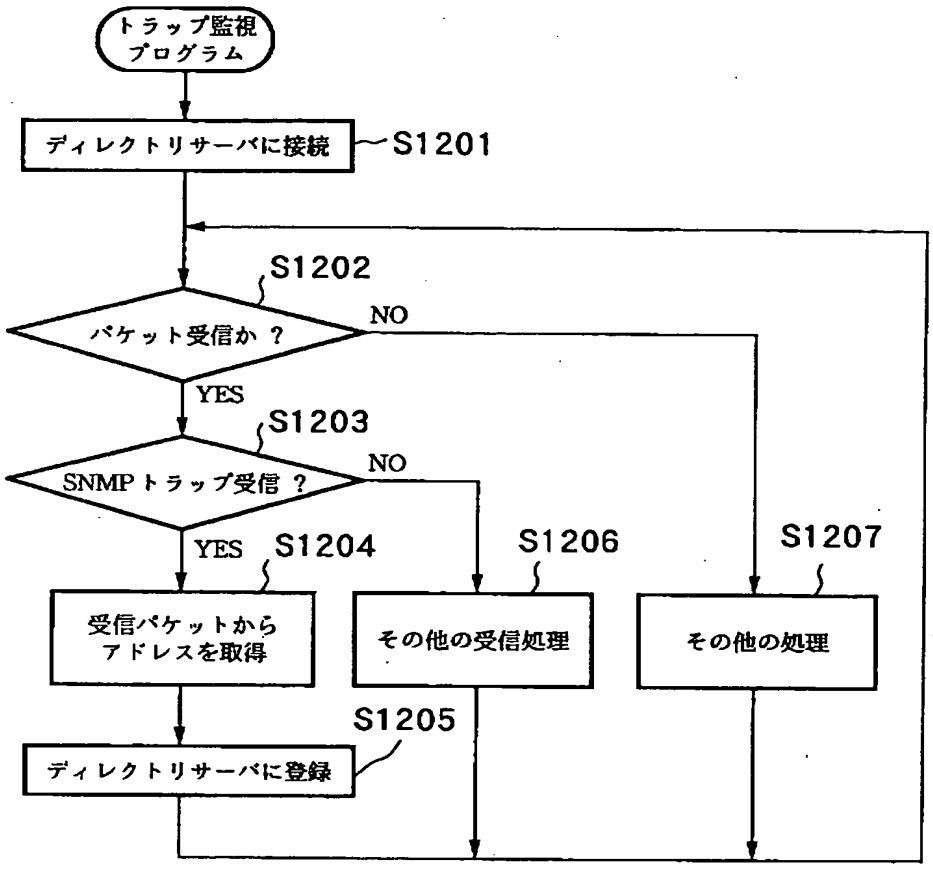
【図１０】


1030
1031
1032
探索モジュール
デバイス１
デバイス２
1020
探索開始
ブロードキャスト
1001
ブロードキャスト
1002
1003
デバイスの応答
1004
一次応答タイマー
1005
デバイスの応答
1006
探索間隔タイマー
1023
デバイス別情報取得（標準MIB）
1007
1008
デバイス応答タイマー
デバイス別情報取得（キャノンMIB）
1009
1010
1021
探索結果通知
1022

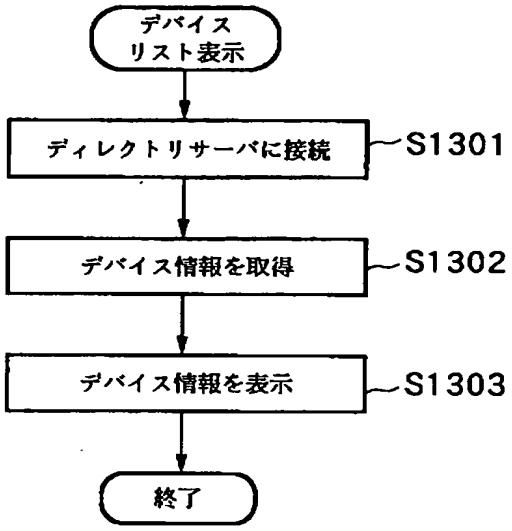
【図１１】


1104
デバイス
1102
LDAPサーバ
DB
1103
トラップ監視プロセス実行マシン
1101
クライアント

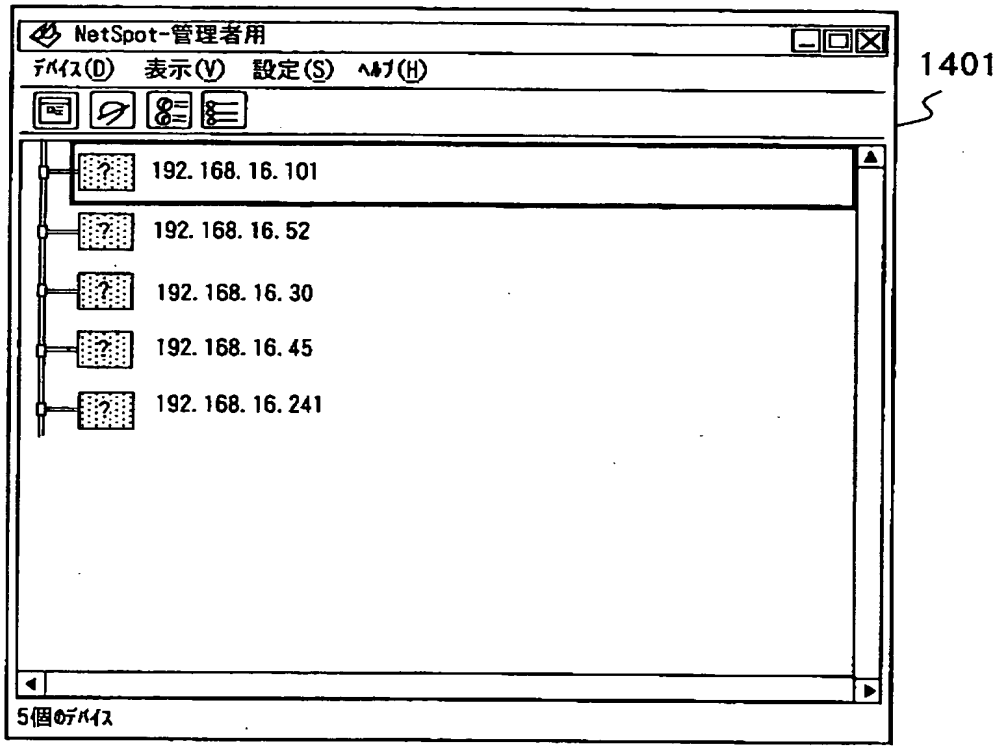
【図１２】


トラップ監視
プログラム
ディレクトリサーバに接続
S1201
S1202
パケット受信か ？
NO
YES
S1203
SNMP トラップ受信 ？
NO
YES
S1204
S1206
S1207
受信パケットから
アドレスを取得
その他の受信処理
その他の処理
S1205
ディレクトリサーバに登録

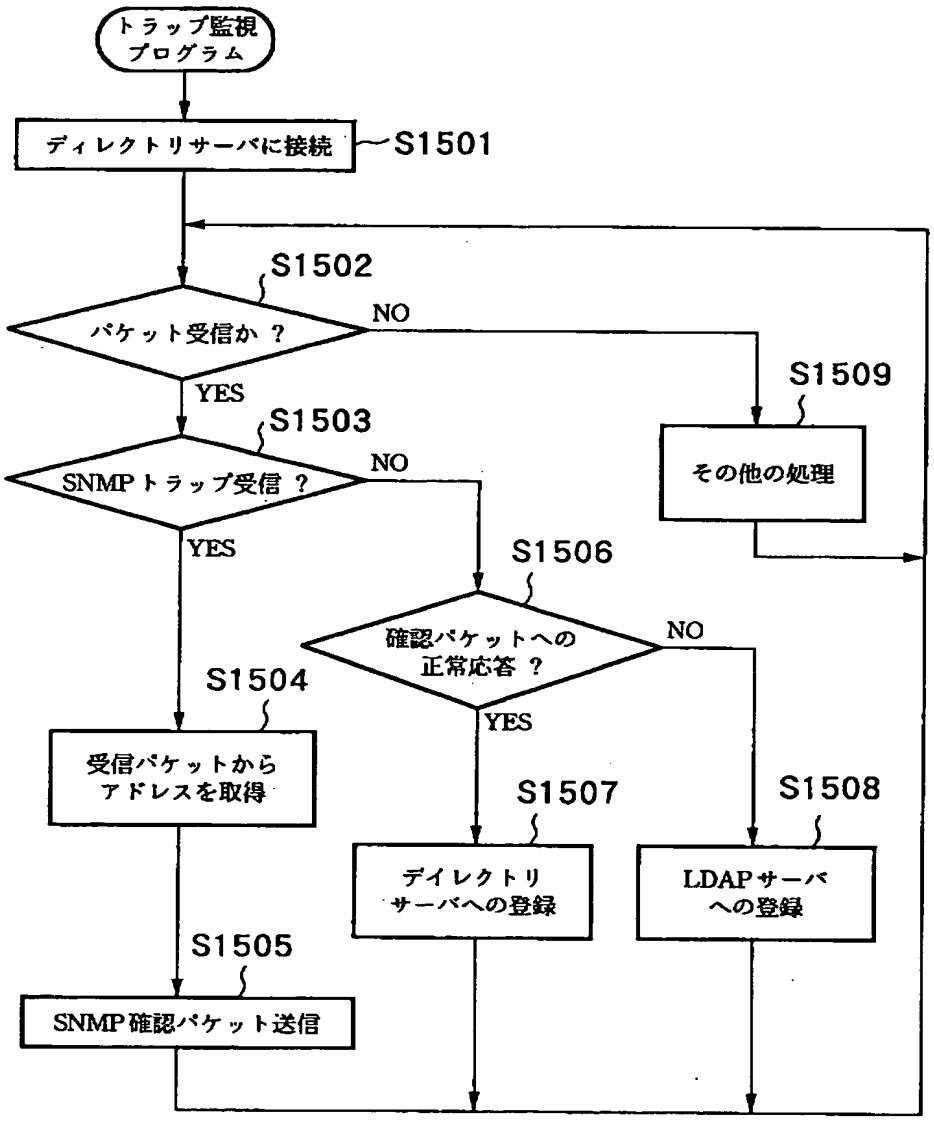
【図１３】


デバイス
リスト表示
ディレクトリサーバに接続
S1301
デバイス情報を取得
S1302
デバイス情報を表示
S1303
終了

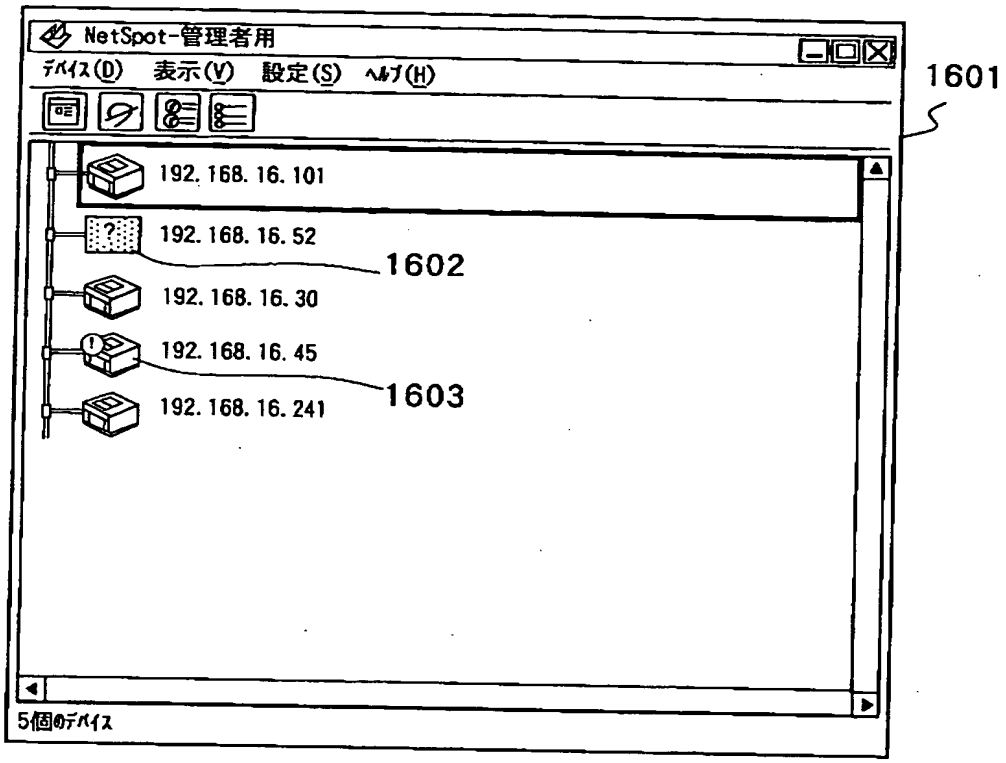
【図１４】


NetSpot-管理者用
ﾃﾞﾊﾞｲｽ(D)
表示(V)
設定(S)
ﾍﾙﾌﾟ(H)
1401
192.168.16.101
192.168.16.52
192.168.16.30
192.168.16.45
192.168.16.241
5個のﾃﾞﾊﾞｲｽ

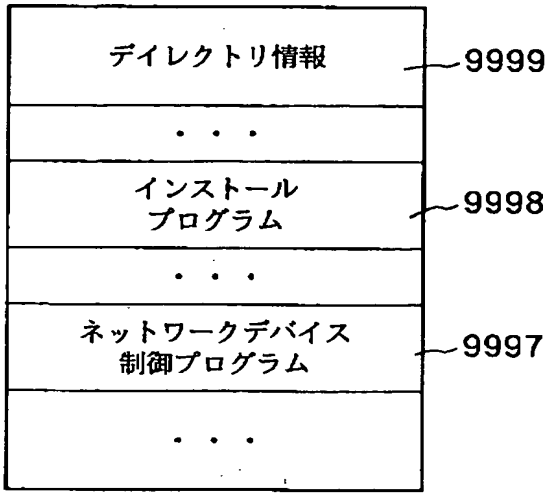
【図１５】


トラップ監視
プログラム
ディレクトリサーバに接続
S1501
S1502
パケット受信か ？
NO
YES
S1509
その他の処理
S1503
SNMPトラップ受信 ？
NO
YES
S1506
確認パケットへの
正常応答 ？
NO
YES
S1504
受信パケットから
アドレスを取得
S1507
デイレクトリ
サーバへの登録
S1508
LDAPサーバ
への登録
S1505
SNMP確認パケット送信

【図１６】


NetSpot-管理者用
ﾃﾞﾊﾞｲｽ(D) 表示(V) 設定(S) ﾍﾙﾌﾟ(H)
192.168.16.101
192.168.16.52
192.168.16.30
192.168.16.45
192.168.16.241
5個のﾃﾞﾊﾞｲｽ
1601
1602
1603

【図１７】



【書類名】　要約書

【要約】

【課題】ネットワークデバイスの探索のためのトラフィックを減らす。

【解決手段】各ネットワークデバイスは、起動時にコールドスタートトラップをブロードキャストする。監視装置はそれを補足し、ＳＮＭＰパケットであるか判定して（Ｓ１２０３）、そうであれば受信パケットからアドレスを取り出してディレクトリサーバに登録する。

【選択図】図１２

出　願　人　履　歴　情　報

識別番号　　〔０００００１００７〕

１．変更年月日　　１９９０年　８月３０日
〔変更理由〕　新規登録
住　所　東京都大田区下丸子３丁目３０番２号
氏　名　キヤノン株式会社